



**Universidad de San Carlos de Guatemala**  
**Facultad de Ingeniería**  
**Escuela de Ingeniería en Ciencias y Sistemas**

**METODOLOGÍAS DE DESARROLLO DE SOFTWARE**  
**APLICADAS EN LA INTEGRACIÓN DE SISTEMAS EN GUATEMALA**

**Griselda Evanidia Quiñonez Reyes**

**Asesorado por el Ingeniero Marlon Antonio Pérez Türk**

**Guatemala, junio de 2009**

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**METODOLOGÍAS DE DESARROLLO DE SOFTWARE  
APLICADAS EN LA INTEGRACIÓN DE SISTEMAS EN GUATEMALA**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA  
FACULTAD DE INGENIERÍA

POR:

**GRISELDA EVANIDIA QUIÑONEZ REYES**

ASESORADO POR EL INGENIERO MARLON ANTONIO PÉREZ TÜRK

AL CONFERÍRSELE EL TÍTULO DE  
**INGENIERA EN CIENCIAS Y SISTEMAS**

GUATEMALA, JUNIO DE 2009

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
FACULTAD DE INGENIERÍA



**NÓMINA DE JUNTA DIRECTIVA**

DECANO	Ing. Murphy Olympo Paiz Recinos
VOCAL I	Inga. Glenda Patricia García Soria
VOCAL II	Inga. Alba Maritza Guerrero de López
VOCAL III	Ing. Miguel Ángel Dávila Calderón
VOCAL IV	Br. José Milton De León Bran
VOCAL V	Br. Isaac Sultán Mejía
SECRETARIA	Inga. Marcia Ivónne Véliz Vargas

**TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO**

DECANO	Ing. Murphy Olympo Paiz Recinos
EXAMINADOR	Ing. César Augusto Fernández Cáceres
EXAMINADOR	Inga. Virginia Victoria Tala Ayerdi
EXAMINADOR	Ing. Edgar Estuardo Santos Sutuj
SECRETARIA	Inga. Marcia Ivónne Véliz Vargas

## **HONORABLE TRIBUNAL EXAMINADOR**

Cumpliendo con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

### **METODOLOGÍAS DE DESARROLLO DE SOFTWARE APLICADAS EN LA INTEGRACIÓN DE SISTEMAS EN GUATEMALA,**

tema que me fuera asignado por la Dirección de la Escuela de Ciencias y Sistemas, en febrero de 2008.

**Griselda Evanidia Quiñonez Reyes**

Guatemala, 21 de marzo de 2009

Señores

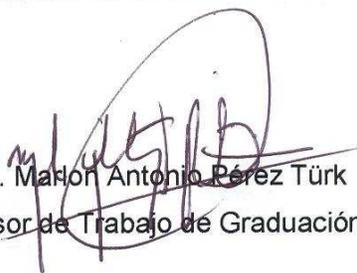
Escuela de Ciencias y Sistemas

Facultad de Ingeniería

Respetables señores:

Atentamente me dirijo a ustedes para hacer de su conocimiento que he revisado el trabajo de graduación titulado: "METODOLOGÍAS DE DESARROLLO DE SOFTWARE APLICADAS EN LA INTEGRACIÓN DE SISTEMAS EN GUATEMALA", realizado por la estudiante GRISELDA EVANIDIA QUIÑONEZ REYES, con carné número 200212397, considerando a mi criterio que dicho trabajo cumple con todos los objetivos propuestos para su desarrollo, según el protocolo.

Sin otro particular, me suscribo de ustedes.



Ing. Mañón Antonio Pérez Türk  
Asesor de Trabajo de Graduación



Universidad San Carlos de Guatemala  
Facultad de Ingeniería  
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 13 de Mayo de 2009

Ingeniero  
**Marlon Antonio Pérez Turk**  
Director de la Escuela de Ingeniería  
En Ciencias y Sistemas

Respetable Ingeniero Pérez:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación del estudiante **GRISelda EVANIDIA QUIÑONEZ REYES**, titulado: "**METODOLOGIAS DE DESARROLLO DE SOFTWARE APLICADAS EN LA INTEGRACION DE SISTEMAS EN GUATEMALA**", y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,

  
**Ing. Carlos Alfredo Azurdia**  
Coordinador de Privados  
y Revisión de Trabajos de Graduación



E  
S  
C  
U  
E  
L  
A  
  
D  
E  
  
C  
I  
E  
N  
C  
I  
A  
S  
  
Y  
  
S  
I  
S  
T  
E  
M  
A  
S

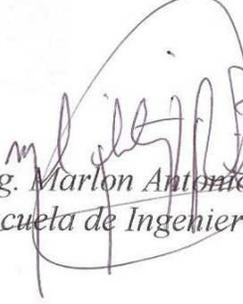
UNIVERSIDAD DE SAN CARLOS  
DE GUATEMALA



FACULTAD DE INGENIERÍA  
ESCUELA DE CIENCIAS Y SISTEMAS  
TEL: 24767644

*El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del asesor con el visto bueno del revisor y del Licenciado en Letras, de trabajo de graduación titulado **“METODOLOGÍAS DE DESARROLLO DE SOFTWARE APLICADAS EN LA INTEGRACIÓN DE SISTEMAS EN GUATEMALA”**, presentado por el estudiante **GRISELDA EVANIDIA QUIÑONEZ REYES**, aprueba el presente trabajo y solicita la autorización del mismo.*

**“ID Y ENSEÑAD A TODOS”**

  
Ing. Marlon Antonio Pérez Turk  
Director, Escuela de Ingeniería Ciencias y Sistemas



Guatemala, 05 de junio 2009

Universidad de San Carlos  
de Guatemala

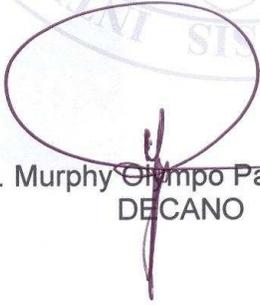


Facultad de Ingeniería  
Decanato

Ref. DTG.199.09

El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería en Ciencias y Sistemas, al trabajo de graduación titulado: **METODOLOGÍAS DE DESARROLLO DE SOFTWARE APLICADAS EN LA INTEGRACIÓN DE SISTEMAS EN GUATEMALA**, presentado por la estudiante universitaria **Griselda Evanidia Quiñonez Reyes**, autoriza la impresión del mismo.

IMPRÍMASE.

  
Ing. Murphy Olimpo Paiz Recinos  
DECANO

Guatemala, junio de 2009



/cc  
c.c. archivo.

## **AGRADECIMIENTOS A:**

### **DIOS**

Por brindarme la sabiduría y el conocimiento necesario, porque ha guiado mi camino y me ha permitido llegar hasta aquí el día de hoy y porque me ha dado fortaleza en los momentos que más lo he necesitado.

### **MI FAMILIA**

A mis papás Gricelda Reyes y Oscar Quiñonez, por su apoyo incondicional y porque siempre me han impulsado para que alcance cada una de las metas de mi vida. A mis hermanos Jorge Alejandro y Oscar Efraín, por su apoyo. A mis abuelitos Pablo Reyes y Consuelo Córdova, por ser mi apoyo en los momentos más difíciles, a mis tíos y primos por todo su cariño y apoyo.

### **MIS AMIGOS**

Víctor Álvarez, Cristian Martínez, Juan José Hernández, Astrid Méndez, Sergio García, Jorge Martínez y Leonel Arrecis, por su amistad, apoyo, paciencia y conocimientos compartidos.

## **ACTO QUE DEDICO A:**

### **DIOS**

Porque me ha dado una vida llena de bendiciones y me ha permitido cumplir mis metas.

### **MIS PADRES**

A la memoria de mi padre Oscar Quiñonez y a mi madre Gricelda Reyes, de quien siempre he recibido todo su amor y apoyo y quien me ha impulsado a alcanzar cada una de mis metas.

### **MIS ABUELITOS**

Con quienes he compartido los momentos más significativos de mi vida y quienes siempre me han acompañado y me han brindado todo su apoyo.

# ÍNDICE GENERAL

<b>ÍNDICE DE ILUSTRACIONES</b>	<b>V</b>
<b>GLOSARIO</b>	<b>VII</b>
<b>RESUMEN</b>	<b>IX</b>
<b>OBJETIVOS</b>	<b>XI</b>
<b>INTRODUCCIÓN</b>	<b>XIII</b>
<b>1. INTEGRACIÓN DE SISTEMAS</b>	<b>1</b>
1.1 Definición	1
1.2 Niveles de integración	1
1.2.1 Nivel 1: Integración punto a punto	2
1.2.2 Nivel 2: Integración estructural	2
1.2.3 Nivel 3: Integración de procesos	2
1.2.4 Nivel 4: Integración externa	3
<b>2. ARQUITECTURAS Y PLATAFORMAS RELEVANTES PARA LA INTEGRACIÓN DE SISTEMAS</b>	<b>5</b>
2.1 <i>Service Oriented Architecture (SOA)</i>	5
2.1.1 Principios fundamentales	5
2.1.2 Ventajas de la integración mediante SOA	7
2.1.3 Riesgos	9
2.1.4 Componentes de SOA	11
<b>3. TECNOLOGÍAS PARA LA INTEGRACIÓN DE SISTEMAS</b>	<b>15</b>
3.1 Tecnología <i>Middleware</i>	15
3.1.1 Definición	15
3.1.2 Tipos de <i>Middleware</i>	15
3.2 EAI: <i>Enterprise Application Integration</i>	19

3.3	BPM: <i>Business Process Management</i>	21
<b>4.</b>	<b>METODOLOGÍAS DE DESARROLLO</b>	<b>23</b>
4.1	Ciclo de desarrollo de software	23
4.1.1	Fases del ciclo de desarrollo de software	23
4.2	Métodos de desarrollo de software	25
4.2.1	Método en cascada	25
4.2.2	Modelo V	27
4.2.3	Modelo de construcción de prototipos	29
4.2.4	Método de espiral	30
4.2.5	<i>Rational Modified Process</i> (RUP)	32
<b>5.</b>	<b>SITUACIÓN ACTUAL DE LOS SISTEMAS DE INFORMACIÓN EN GUATEMALA</b>	<b>37</b>
5.1	Introducción	37
5.2	Detalle de la encuesta y resultados	38
5.2.1	Sector de la organización	38
5.2.2	Tamaño de la organización	39
5.2.3	Rol de la organización	40
5.2.4	Tipo de organización	44
5.2.5	Sistemas heterogéneos	45
5.2.6	¿Están integrando sus sistemas las empresas?	47
5.2.7	Plataformas de desarrollo	49
5.2.8	Herramientas para administración de información	51
5.2.9	¿Cómo integrarán sus sistemas las empresas?	53
5.2.10	Metodologías para integrar sistemas de información	55
5.2.11	Niveles de integración de los sistemas de información	57
5.2.12	Tecnologías utilizadas para la integración de sistemas de información	60
5.2.13	Arquitectura de los sistemas integrados	62
5.2.14	Herramientas más utilizadas para la integración de sistemas	64

<b>6. TENDENCIAS Y EXPECTATIVAS EN LA INTEGRACIÓN DE SISTEMAS DE INFORMACIÓN EN GUATEMALA</b>	<b>67</b>
6.1 Tendencias	67
6.2 Expectativas	69
<b>CONCLUSIONES</b>	<b>71</b>
<b>RECOMENDACIONES</b>	<b>73</b>
<b>BIBLIOGRAFÍAS</b>	<b>75</b>



# ÍNDICE DE ILUSTRACIONES

## FIGURAS

1	Componentes de SOA	11
2	Arquitectura accidental típica	20
3	Arquitectura <i>Hub and Spoke</i>	21
4	Etapas del método en cascada	25
5	Etapas del modelo V	28
6	Etapas del modelo de prototipos	30
7	Fases del método en espiral	31
8	Integración del trabajo	33
9	Evolución de la arquitectura en las fases de RUP	34
10	Iteración en RUP	35
11	Esfuerzo en cada fase de RUP	36
12	Organizaciones por sector	38
13	Organizaciones por sector (en proporciones)	39
14	Organizaciones por tamaño	40
15	Organizaciones por tamaño (en proporciones)	40
16	Proporciones de roles de organizaciones	42
17	Tipo de organizaciones	44
18	Tipo de organizaciones (en proporciones)	45
19	Sistemas heterogéneos por sector	46
20	Proporción de organizaciones que poseen sistemas heterogéneos	46
21	Integración de sistemas en Guatemala	48

22	Integración de sistemas heterogéneos (proporciones)	49
23	Plataformas de desarrollo utilizadas en los sistemas de información en Guatemala	50
24	Plataformas de desarrollo utilizadas (en proporciones)	51
25	Herramientas para administración de la información	52
26	Herramientas para administrar información (proporciones)	53
27	Modalidad para desarrollo de sistemas en Guatemala	54
28	Modalidad para integrar sistemas de información (en proporciones)	55
29	Metodologías de desarrollo utilizadas en sistemas de información en Guatemala	56
30	Metodologías de desarrollo (en proporciones)	57
31	Niveles de integración de sistemas en Guatemala	59
32	Niveles de integración de sistemas en Guatemala (en proporciones)	59
33	Tecnologías utilizadas para la integración de sistemas en Guatemala	61
34	Tecnologías utilizadas para la integración de sistemas en Guatemala (proporciones)	62
35	Arquitectura de sistemas integrados	63
36	Arquitectura de sistemas integrados (proporciones)	63
37	Herramientas para la integración de sistemas	65

## **TABLAS**

I	Roles de las organizaciones	41
II	Actividades de la población encuestada (en porcentajes)	43

## GLOSARIO

<b>API</b>	<i>Application Program Interface</i> . Conjunto de rutinas del sistema que se pueden utilizar en un programa para la gestión de datos entrada y salida, gestión de ficheros, etc.
<b>Aplicación</b>	Programa que realiza una serie de funciones y que permite al usuario interactuar con el ordenador.
<b>Base de datos</b>	Almacenamiento colectivo de las bibliotecas de datos que son requeridas y organizadas para cubrir los requisitos de procesos y recuperación de información.
<b>Baseline</b>	Línea base. Es una “imagen congelada” de los elementos de configuración, tales como software, documentación, hardware, de una versión o revisión determinada, de una fecha específica, que define la configuración del producto. Provee una base para la administración de cambios.
<b>Caso de uso</b>	Diagrama que representa una estructura utilizada para describir la forma en que un sistema luce para un usuario.

## **VSAM**

***Virtual Storage Access Method (VSAM)*** es un esquema de almacenamiento de IBM del sistema operativo OS/VS2. Es un sistema de ficheros orientado a registros que pueden estar organizados de cuatro maneras diferentes: *Key Sequenced Data Set (KSDS)*, *Relative Record Data Set (RRDS)*, *Entry Sequenced Data Set (ESDS)* y *Linear Data Set (LDS)*. Mientras los tipos KSDS, RRDS y ESDS contienen registros, el tipo LDS (añadido después a VSAM) contiene una secuencia de bytes sin ningún orden de organización intrínseco.

## RESUMEN

Un aspecto importante que debe tomarse en cuenta es la evolución de los sistemas de información y las tendencias que van presentando en cuanto a las necesidades que cubren, arquitecturas que utilizan, metodologías para su desarrollo e integración, entre otros aspectos.

Con el avance de la tecnología, las opciones que ofrece el mercado para integrar sistemas facilitan cada vez más el trabajo de integración y permiten incorporar sistemas cada vez más complejos.

Es por ello, que se hace necesario conocer los aspectos que abarca la integración de sistemas de información, el proceso como tal y los niveles en los que se puede catalogar cada uno de los sistemas, basados en características propias que presenta el proceso de interacción de aplicaciones.

Lo que respecta a la arquitectura de sistemas de información, es de mucha utilidad conocer acerca de los modelos que facilitan la integración y también las plataformas que trabajan conjuntamente con estas arquitecturas, que conforman un todo, proveyendo servicios y estableciendo estándares.

También es relevante conocer las metodologías que pueden ser utilizadas y aplicadas dependiendo de la magnitud de cada sistema y como escoger la más adecuada para su desarrollo.



# OBJETIVOS

## General

Conocer las tecnologías, herramientas y aplicación de las distintas metodologías de desarrollo de software más utilizadas para la integración de sistemas de información y los niveles de integración que presentan los sistemas heterogéneos en las empresas guatemaltecas.

## Específicos:

1. Conocer las distintas metodologías utilizadas por empresas guatemaltecas para desarrollar e integrar sistemas de información.
2. Investigar las metodologías utilizadas en empresas guatemaltecas para afrontar la integración de sus sistemas.
3. Dar a conocer el proceso de integración de sistemas de información y la forma en la que éste proceso se lleva a cabo, a través de qué metodologías y herramientas.
4. Establecer tendencias y expectativas en la integración de sistemas en Guatemala.



## INTRODUCCIÓN

A través del presente trabajo, se plantean aspectos importantes relacionados con la tecnología de la información como el desarrollo y la integración de sistemas de información.

Cabe mencionar que actualmente existen diversas herramientas para la integración de los distintos sistemas de información, lo que permite que en una institución exista más de un sistema y que pueda interactuar con otros sistemas sin importar que no tengan características en común, como por ejemplo, la plataforma y los lenguajes de programación.

Pero además de las herramientas, también existen arquitecturas que facilitan la integración de los sistemas; es por ello que en este trabajo se ha abarcado el tema de arquitecturas específicas como lo es SOA o arquitectura orientada a servicios, que de acuerdo a las tendencias que presentan todas las organizaciones que desarrollan e integran sistemas, ha sido uno de los paradigmas de arquitecturas a los que cada vez más organizaciones se integran, es un estándar que se ha ido estableciendo para facilitar la interacción entre sistemas heterogéneos de información, permitiendo que la integración no solamente sea a nivel interno de una organización sino que también sea a nivel externo con sistemas de otras organizaciones, definiendo aspectos como estos, el nivel de integración en el que se encuentra una institución.

Además de conocer las arquitecturas y las herramientas, un punto importante es la manera en la que se desarrollan todos estos sistemas, dado

que, debido a la complejidad de las funciones que deben realizar los sistemas informáticos, así deberá seleccionarse la metodología de desarrollo que más se adapte y permita obtener los resultados esperados, es por ello que también existe un apartado que ofrece una breve introducción acerca de las metodologías de desarrollo más comunes donde se destacan sus ventajas y desventajas y características de los proyectos que son claves para decidir trabajar con una u otra metodología.

# **1. INTEGRACIÓN DE SISTEMAS**

## **1.1 Definición**

Integración de sistemas es un término utilizado para definir la unión coherente de actividades o procesos diferentes; relaciona estrechamente el negocio y los sistemas de información, tanto que la integración de una parte es necesaria para que pueda funcionar la otra.

La integración del negocio es la creación de actividades de negocio coordinadas, administradas por diferentes personas, en grupo o por organizaciones, de tal forma que se pueda formar un proceso de negocio unificado. Para que el negocio pueda estar integrado, es necesario contar con sistemas de información integrados, capaces de combinar informaciones provenientes de diversas fuentes.

La integración de sistemas de información reduce la duplicidad de esfuerzos y facilita la interacción de las unidades de la organización.

## **1.2 Niveles de integración**

Los niveles de integración están orientados a las Tecnologías de Información; representan el grado de cooperación mutua entre actividades o procesos distintos. La diferencia de un nivel a otro representa la evolución de la empresa en el tema de integración, permitiendo definir donde se encuentra

actualmente, hasta donde quiere llegar y la estrategia a utilizar para alcanzar el nivel establecido.

### **1.2.1 Nivel 1: Integración punto a punto**

Es el escalón inicial de los niveles de integración. En este nivel los sistemas son altamente independientes, solamente se define una infraestructura básica para intercambiar datos entre aplicaciones y se utilizan interfaces intermedias que no contemplan la visión del negocio.

### **1.2.2 Nivel 2: Integración estructural**

En este nivel se incorpora la tecnología *Middleware*, lo que permite estandarizar y controlar el intercambio de información. Presenta las siguientes características:

- a. Una estructura central que controla el intercambio de información.
- b. Las transacciones entre aplicaciones y las reglas del negocio las consolida el *Middleware*.
- c. No integra componentes externos del negocio.
- d. Fuentes de datos comunes e interfaces integradas.

### **1.2.3 Nivel 3: Integración de procesos**

En este nivel se consideran las reglas del negocio y el flujo de procesos; es por ello que la empresa ya cuenta con un modelo de negocio común,

administra el flujo de información entre aplicaciones e incorpora tecnología sofisticada para en la capa *Middleware*.

#### **1.2.4 Nivel 4: Integración externa**

Es el nivel más alto en la escala de integración. En este nivel las empresas están enfocadas en la mejora continua del servicio al cliente, para ello buscan redefinir la organización considerando las aplicaciones empresariales, la influencia tecnológica, la transformación de los procesos de negocio y las nuevas estructuras. Presenta las siguientes características:

- a. Utilización de tecnologías EAI para transformar el negocio, permitiendo unir directamente las operaciones internas entre clientes y proveedores.
- b. Contempla nuevas capacidades para crear e innovar productos y servicios en línea.
- c. Tiene la capacidad de mejorar una marca registrada o crear una nueva identidad en Internet



## 2. ARQUITECTURAS Y PLATAFORMAS RELEVANTES PARA LA INTEGRACIÓN DE SISTEMAS

### 2.1 *Service Oriented Architecture (SOA)*

La Arquitectura Orientada a Servicios (SOA por sus siglas en inglés), es una arquitectura cuyo objetivo es crear una malla de servicios disponibles a nivel mundial y establecer guías para integrar diferentes sistemas informáticos. Es un paradigma de arquitectura de sistemas que consiste en identificar, racionalizar y exponer servicios existentes en una unidad organizacional para su posterior reutilización; así como coordinar los servicios publicados para realizar funcionalidades más complejas.

Un servicio se puede definir como una aplicación que utiliza el intercambio de mensajes bien definidos para la interacción; el valor de negocio que tiene un servicio es proporcionado por distintos sistemas.

#### 2.1.1 Principios fundamentales

Tenemos cuatro principios que definen el estilo de arquitectura orientado a servicios, los cuales nos ayudarán y orientarán en el proceso de desarrollo. Estos principios son:

- **Los límites son explícitos.** Los servicios publican un contrato para definir las interfaces públicas que ofrecen. Para facilitar su consumo estos

interfaces deben ser simples y deben ocultar los detalles de su implementación.

- **Los servicios son autónomos.** Cada servicio tiene su propia implementación, distribución y entorno operativo, por consiguiente se versionan y administran de forma independiente. Sin embargo aunque los servicios están diseñados para ser autónomos, ningún servicio es completamente independiente. Las soluciones basadas en SOA están formadas por una serie de servicios configurados para una solución específica
- **Los servicios comparten esquema y contrato, pero no implementación.** El mayor reto de este principio es hacer que estos esquemas y contratos no cambien una vez que se han publicado, evitando así el mínimo impacto en sus usuarios. La línea que separa los datos internos de los externos es de vital importancia para la implementación y el uso de un servicio determinado. Los datos externos o públicos deben estar definidos por estándares, mientras que los internos o privados deben encapsularse en el interior de la implementación del servicio.
- **Colaboración de servicios.** Se refiere al conjunto de interacciones necesarias entre distintos servicios para crear una funcionalidad única. Los distintos tipos de colaboración son:

#### **a. Orquestación**

Tipo de colaboración en la que existe un servicio principal que invoca directamente a otros servicios; el servicio principal conoce los servicios que invoca: acciones, interfaces, respuestas y resultados.

## **b. Interacción de negocio**

Es un tipo de colaboración que cuenta con la interacción del negocio e involucra mensajes de varias partes. Es un tipo de mecanismo de coordinación que conoce: acciones, posibles estados y caminos de interacción entre servicios. Este mecanismo puede ser: un manejador de eventos, un motor de procesos, un motor de *workflow* o un *Enterprise Service Bus (ESB)*

## **c. Intercepción**

Es un tipo de colaboración en la que existe un servicio intermediario que recibe y actúa sobre un pedido o respuesta y luego lo reenvía al destinatario original. Este tipo de colaboración es utilizada para realizar funciones comunes y transversales al negocio, tales como: seguridad, política, auditoría y traducción.

### **2.1.2 Ventajas de la integración mediante SOA**

Una de las ventajas principales de una estrategia de integración de este tipo es la comunicación de manera automática (sin intervención humana) de todas y cada una de las aplicaciones con las que contamos y que son necesarias en cada proceso de la empresa.

#### **2.1.2.1 Débil acoplamiento**

Como es de esperar esta ventaja no es exclusiva de este estilo de arquitectura, otros como la integración a través de bases de datos consiguen el

mismo objetivo. Sin embargo uno de los más importantes éxitos que consiguen la orientación a servicios es el débil acoplamiento entre los distintos sistemas.

Esta característica permite diseñar un sistema empresarial más ágil y adaptable a los cambios que en un momento dado pueda necesitar la compañía. El débil acoplamiento entre sistemas permite que estos puedan ser intercambiados sin necesidad de realizar grandes cambios en el resto de componentes del sistema, hasta el punto de que el resto de componentes ni siquiera perciban el cambio o evolución de uno de ellos.

#### **2.1.2.2 Asincronía**

Otra gran ventaja que podemos obtener de la arquitectura orientada a servicios SOA es la comunicación asíncrona entre las diferentes aplicaciones. Esto significa principalmente que cuando un sistema solicita a otro un determinado procesamiento, no es necesario que el primero de ellos quede bloqueado a la espera de una respuesta.

Esta característica es muy importante en un sistema empresarial tan heterogéneo donde la complejidad de las operaciones difiere entre unos sistemas y otros. Además, permitimos que cada aplicación trabaje a su ritmo sin depender de la velocidad con la que trabaja el resto de sistemas, consumiendo peticiones en la medida de sus posibilidades y no bloqueándose porque otra funcione más despacio.

Otro aspecto a destacar es la robustez del sistema ante la caída de uno de sus componentes.

### **2.1.2.3 Reutilización**

Si durante el diseño de la arquitectura seguimos fielmente los principios fundamentales definidos en este estilo de arquitectura, es decir, definimos un contrato formal estable para nuestros servicios, implantamos los servicios tal que funcionen con plena autonomía y por último permitimos que puedan ser fácilmente descubiertos, la consecuencia directa de este diseño será una reutilización óptima de los servicios desarrollados.

### **2.1.3 Riesgos**

#### **2.1.3.1 Difícil definición del nivel de detalle de los servicios**

La difícil conceptualización e implementación de los servicios es uno de los problemas principales. El concepto no es sencillo y aun cuando se ha comprendido es relativamente sencillo cometer errores en el diseño de cada uno de los servicios.

Un error muy común cuando exponemos la funcionalidad de un sistema a través de servicios, es elegir el tamaño del grano de estos servicios. Esto quiere decir, que si elegimos un grano muy fino es muy probable que el rendimiento de del sistema caiga rápidamente. Esto es debido a que para ejecutar una operación de más envergadura necesitaremos llamar a muchos pequeños procedimientos, que al encontrarse desacoplados e intermediados por diferentes capas de responsabilidad necesarias para implementar este tipo de comunicaciones, la complejidad del procesamiento aumenta bastante.

Tampoco podemos irnos al otro extremo y definir servicios con un grano demasiado grande, ya que en ese caso corremos el riesgo de diseñar cada servicio de manera tan específica a la necesidad concreta del proceso de negocio, que hará muy complicado reutilizar dicho servicio por otra aplicación en otro proceso de negocio de la empresa. Esta situación nos condenaría a desarrollar servicios a medida para cada cliente lo que no ayudará a la agilidad y adaptabilidad que estamos buscando.

### **2.1.3.2 Necesidades de monitorización**

A toda esta complejidad, debemos añadir la necesidad no funcional de desarrollar procedimientos que permitan monitorizar el funcionamiento del sistema *Middleware*. Esto es muy importante ya que en sistemas de cierta envergadura, si sólo nos preocupamos por controlar el rendimiento de los sistemas finales, podemos llevarnos la sorpresa de estar perdiendo información o que simplemente comience a degradarse hasta provocar el colapso del sistema de integración.

### **2.1.3.3 Diferentes modelos de información**

Por último, señalaremos otra gran preocupación de las arquitecturas SOA y de las arquitecturas de integración en general. Es el hecho que con toda seguridad cada sistema va a manejar un modelo de información con distinto formato y contenido.

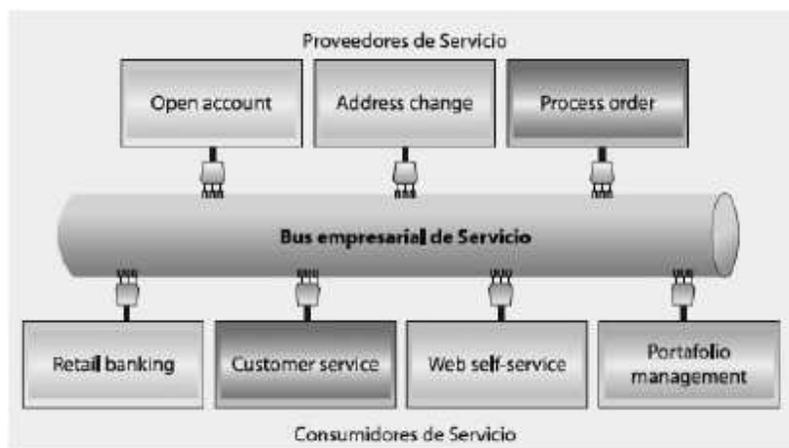
Esta circunstancia obliga a nuestro *middleware* no solo a comunicar unos sistemas con otros sino hacerlo de manera que se entiendan. Por consiguiente también entra dentro del dominio de SOA el tratamiento y transformación de la información para que esta sea legible en su destino.

## 2.1.4 Componentes de SOA

Los principales componentes son tres: los sistemas proveedores de servicios, los sistemas clientes o consumidores de servicios y un bus de servicios que canalizará todos los flujos de información entre los proveedores de servicios y los clientes de servicios.

La adopción de una arquitectura basada en servicios requiere de una infraestructura de comunicaciones escalable y segura entre los componentes. Esto es lo que se conoce como *Enterprise Service Bus* (Bus Empresarial de Servicio).

**Figura 1. Componentes de SOA**



### 2.1.4.1 Proveedores de servicios

Un proveedor de servicios es una entidad de software que implementa una especificación de servicio. Existen dos tipos específicos de proveedores de servicios, estos son:

- *Service locator*: Registra los servicios y permite buscar interfaces de proveedores de servicios y sus ubicaciones.
- *Service broker*: Pasa requerimientos de servicios a otros proveedores de servicios.

#### **2.1.4.2 Consumidores de servicios**

Entidad de software que llama a un proveedor de servicios. Tradicionalmente se le llama “cliente”. Puede ser una aplicación final u otro servicio.

#### **2.1.4.3 *Enterprise Service Bus (ESB)***

Para poder adoptar una arquitectura orientada a servicios, se recomienda la utilización de una plataforma tecnológica que proporcione los mecanismos adecuados para asegurar que los principios en los que se basa la arquitectura orientada a servicios tengan un reflejo en la infraestructura tecnológica de la entidad; para ello, se ha consolidado una nueva categoría de software conocida como *Enterprise Service Bus (ESB)*, que se define como un bus de comunicación o mensajería uniforme entre servicios, tiene las siguientes características:

- Facilita la definición de interfaces visibles y comprensibles por el resto de servicios, ya que soporta la definición de servicios sin importar la plataforma tecnológica real. A su vez, resuelve la mediación entre interfaces y sistemas reales que proporcionan el servicio y también incorpora las capacidades de registro de todos los servicios disponibles en el bus, todo esto conlleva a que se pueda contar con

una visión unificada y consistente en cuanto a los servicios de una entidad que pueden ser utilizados en cualquier momento.

- Permite resolver heterogeneidades en los mensajes que manejan los distintos servicios, ya que parte integral del bus es la inclusión de servicios de transformación distribuida de datos.
- Independientemente del tipo de sistema que proporcione un servicio, este puede ser incorporado al bus (alta extensibilidad).
- Con base a la definición de procesos complejos que involucran los servicios disponibles en el bus y el contenido de mensajes, define un enrutamiento inteligente.
- Utiliza mecanismos para mejorar la calidad de los servicios que ofrece además de garantizar su disponibilidad, distribución e integridad transaccional cuando un proceso requiere la utilización de varios servicios.
- Tiene la capacidad de auditar el estado y los servicios disponibles en el bus a nivel tecnológico.



### 3. TECNOLOGÍAS PARA LA INTEGRACIÓN DE SISTEMAS

#### 3.1 Tecnología *Middleware*

##### 3.1.1 Definición

*Middleware* se define como la capa de software que se coloca entre el usuario y el entorno distribuido, abstrayendo al usuario de la complejidad y la heterogeneidad de las arquitecturas, protocolos, sistemas operativos, lenguajes de programación; es decir, otorga la posibilidad de intercomunicar aplicaciones desarrolladas en distintos lenguajes de programación, sistemas operativos y plataformas.<sup>1</sup>

##### 3.1.2 Tipos de *Middleware*

- ***Distributed Tuples (DT)***. Orientado al acceso a base de datos. Permite el desarrollo de sistemas independientes del manejador de base de datos que lo soporta.
- ***Remote Procedure Call (RPC)***. Permite la gestión remota de redes ya que es diseñado como servicio síncrono. Funciona de la siguiente forma: realiza la llamada y pasa el control al procedimiento servidor, al momento de tener

---

<sup>1</sup> Orientaciones para la selección de tecnologías de integración de sistemas de software.  
[http://www.lisi.usb.ve/publicaciones/07%20integracion%20de%20sistemas/integracion\\_03.pdf](http://www.lisi.usb.ve/publicaciones/07%20integracion%20de%20sistemas/integracion_03.pdf)

un resultado el cliente recupera el control. Se ocupa de tres tareas: interfaz del servicio, búsqueda de servidor y gestión de comunicación.

- ***Messaging Oriented Middleware (MOM)***. Orientado a la administración de colas y envío de mensajes entre aplicaciones, está diseñado con tecnología asíncrona. El servidor y el cliente se ejecutan de manera asíncrona por lo que no se requiere respuesta; las aplicaciones únicamente colocan y sacan mensajes de las colas.
  
- ***Distributed Object Middleware (DOM)***. Definido para las tecnologías orientadas a objetos. Su núcleo es el *Object Request Broker (ORB)* y su finalidad es establecer comunicación entre clientes y objetos de forma transparente (objetos que solicitan servicio a otros objetos que se encuentran en la red).
  
- ***Common Object Request Broker Architecture (CORBA)***. Es un modelo mediante el que se logra la interacción de objetos heterogéneos en protocolos de comunicación, plataformas y lenguajes de desarrollo. Está orientado a sistemas distribuidos. Está basado en:
  - a. Un lenguaje para describir la interfaz de un objeto: *Interface Definition Language (IDL)*.
  
  - b. Complementos para objetos utilizados en el desarrollo de aplicaciones: Servicios CORBA
  
  - c. Cobertura de servicios de alto nivel, tales como redes, interfaces y administración de sistemas: facilidades CORBA.

- d. Proveer funcionalidad en áreas específicas: interfaces de dominio CORBA.
- e. Protocolo para empaquetar y definir mensajes transmitidos entre objetos:  
*General Inter-ORB Protocol (GIOP)*
- **Remote Method Invocation (RMI).** Orientado, al igual que RPC, a la gestión remota de redes, con la diferencia que es un *Middleware* soportado por Java (lenguaje orientado a objetos). Incorpora máquinas virtuales en las que residen los servicios y que pueden ser invocados desde otras máquinas virtuales distintas a donde reside el cliente.
- **Distributed Component Object Model (DCOM).** Especializado en la plataforma Windows, permite la interacción de objetos entre las distintas versiones de sistemas operativos Windows.
- **Transaction Processing Monitors (TP Monitors).** Orientado al procesamiento de transacciones, su objetivo principal es la coordinación solicitudes entre dispositivos y aplicaciones que procesan solicitudes. Facilita la conectividad y el acceso a gran número de usuarios con servicios de *back-end* limitados. Debido a su función principal, requiere de un monitor de transacciones entre procesos para llevar el control y garantizar transacciones exitosas, así como las acciones que debe tomar en las transacciones fallidas.
- **Database Access Technology (DBAT).** Se llama así a todas las Interfaces de Programación de Aplicaciones (API por sus siglas en inglés) a través de las que se crea una capa para el acceso a bases de datos. Estas interfaces son librerías que ocultan la complejidad del manejador de base de datos y

que son utilizadas por otros programas como una capa de abstracción. Entre las más conocidas se pueden mencionar:

- *Java DataBase Connectivity* (JDBC). Es una API utilizada para manejar las conexiones y operaciones a bases de datos desde el lenguaje de programación Java. A nivel de base de datos, maneja el Lenguaje Estructurado de Consultas (SQL) y es a través de este lenguaje que permite al usuario administrar bases de datos y dar mantenimiento a la información, según sean los permisos del usuario que se conecta a la base de datos.
- *Open Database Connectivity* (ODBC). Es una interfaz abierta utilizada para el acceso y la realización de operaciones en bases de datos, independientemente del sistema manejador de base de datos; esto es que cualquier aplicación puede acceder y operar sobre cualquier base de datos siempre que se utilicen sentencias SQL.
- ***Component Oriented Framework (COF)***. Orientado y soportado por el modelo de desarrollo de aplicaciones basado en componentes, siendo los componentes los que permiten realizar la integración entre aplicaciones, incluyendo con ello también la reusabilidad.
- ***Directory Services (DS)***. Este tipo de *middleware* está basado en directorios. Utiliza una combinación única de acceso y contraseña; para acceder utiliza un protocolo específico llamado *Lightweight Directory Access Protocol* (LDAP). También aumenta la seguridad y proporciona una localización única para la información de los usuarios.

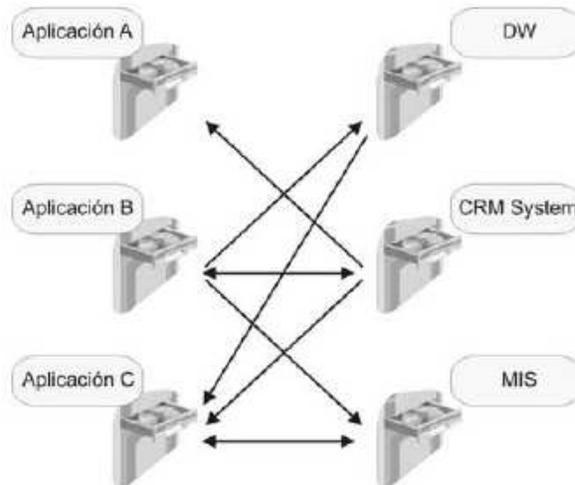
- **Application Servers (AS).** Son servidores que se encuentran en una red de computadoras, encargados de proporcionar servicios de aplicación a computadoras cliente; generalmente encargados de las funciones de lógica del negocio, proveen APIs para las conexiones y operaciones sobre bases de datos y soportan estándares como HTML, XML, SSL, entre otros, que les permiten funcionar en ambientes Web. Entre las ventajas que presentan se pueden mencionar la centralización y disminución en la complejidad en el desarrollo de aplicaciones.

### 3.2 EAI: *Enterprise Application Integration*

Es un concepto que engloba varios aspectos necesarios para la conexión de sistemas, datos y procesos de una o varias entidades, tales como: herramientas, tecnologías, procesos y metodologías. Se denomina integración *Business to Business* (B2Bi) cuando se conectan varias entidades diferentes.

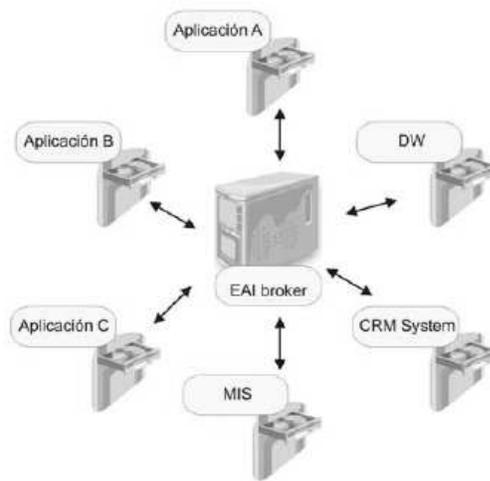
Inicialmente el proceso para realizar la integración entre sistemas se realizó a través de conexiones punto a punto, que involucraban cambios en los sistemas para lograr la comunicación pero que a su vez tenían las desventajas de poner en riesgo la integridad y complicar el mantenimiento de los mismos, además de hacer muy difíciles las tareas de monitoreo de comunicación, control de dependencias y abordar nuevas necesidades de integración de una forma ágil; esta integración utiliza lo que se denomina arquitectura accidental.

**Figura 2. Arquitectura accidental típica**



Con el propósito de reducir los problemas que presentaba la arquitectura accidental, surge un tipo de software que busca facilitar y controlar la integración de sistemas heterogéneos llamado *broker* de integración (EAI *broker*). Este software utiliza una arquitectura *hub-and-spoke*. En esta arquitectura, el *broker* es el intermediario para comunicar los sistemas, logrando con esto reducir el número de conexiones necesarias, además de proporcionar conectividad con diferentes aplicaciones, motores que transforman mensajes permitiendo ajustarlos al formato que espera el destino y tecnologías a través de adaptadores.

**Figura 3. Arquitectura *Hub and Spoke***



A pesar de todas las ventajas que presentan los EAI *brokers* respecto a la arquitectura accidental, tienen desventajas tales como tener un alto coste de implantación además de ser excesivamente rígidos en las integraciones resultantes.

### **3.3 BPM: *Business Process Management***

El Software de Gestión de Procesos de Negocio (BPM por sus siglas en inglés) surge como una evolución de los EAI *brokers*, ya que cubre una de sus mayores deficiencias: la especificación de procesos desde el punto de vista de negocio.

Las mejoras presentadas por este producto respecto los EAI *brokers* son:

- Procesos definidos con mayor énfasis en las reglas del negocio, logrando con ello que sean más apegados a las necesidades reales de la integración.

- Priorizar el uso de estándares para facilitar la interoperabilidad y creación de sistemas abiertos, con la finalidad de abandonar formatos propietarios de forma progresiva.
- Mejora, desde el punto de vista del negocio, de la evaluación, análisis y monitoreo de procesos.
- Reducción de varios factores tales como la necesidad de escribir código ya que la integración se realiza en base al diseño de procesos y respecto a los EAI *brokers*, se puede mencionar la reducción en el coste de integración y la curva de aprendizaje.

## **4. METODOLOGÍAS DE DESARROLLO**

### **4.1 Ciclo de desarrollo de software**

Este concepto define de manera estándar las fases involucradas en el desarrollo de cualquier proyecto de software, independientemente de la metodología y los términos adoptados por cada una, el propósito de cada fase es el mismo.

#### **4.1.1 Fases del ciclo de desarrollo de software**

##### **4.1.1.1 Fase de concepto**

En esta fase se define el problema que el software resolverá lo que permite evaluar el tamaño y alcance del proyecto; también se determinan los criterios para considerar el éxito o fracaso del mismo.

##### **4.1.1.2 Fase de especificación**

Es la fase donde el equipo de tecnología conoce los problemas que el usuario busca resolver con el producto que se va a desarrollar y las diferentes facetas de los procesos del negocio, en otras palabras, es en esta fase donde se detalla cada requerimiento a desarrollar.

#### **4.1.1.3 Fase de diseño**

Es la fase donde se evalúa y planifica la fase de construcción del proyecto; además se toman las decisiones técnicas en cuanto a la arquitectura que se implementara para el proyecto de acuerdo con la dirección de la arquitectura de la empresa.

#### **4.1.1.4 Fase de construcción**

Es la fase de desarrollo del producto, donde se define el ambiente de desarrollo, compilación y construcción de componentes, que en conjunto todo esto es lo que se denomina el desarrollo de código.

#### **4.1.1.5 Fase de aseguramiento de la calidad**

El software al igual que todo producto previo a entregarse al usuario final, también pasa por una etapa de aseguramiento de la calidad, para ello es importante medir factores como: satisfacción del requerimiento, desempeño del sistema y la facilidad de uso para el usuario, entre otros.

#### **4.1.1.6 Fase de implementación**

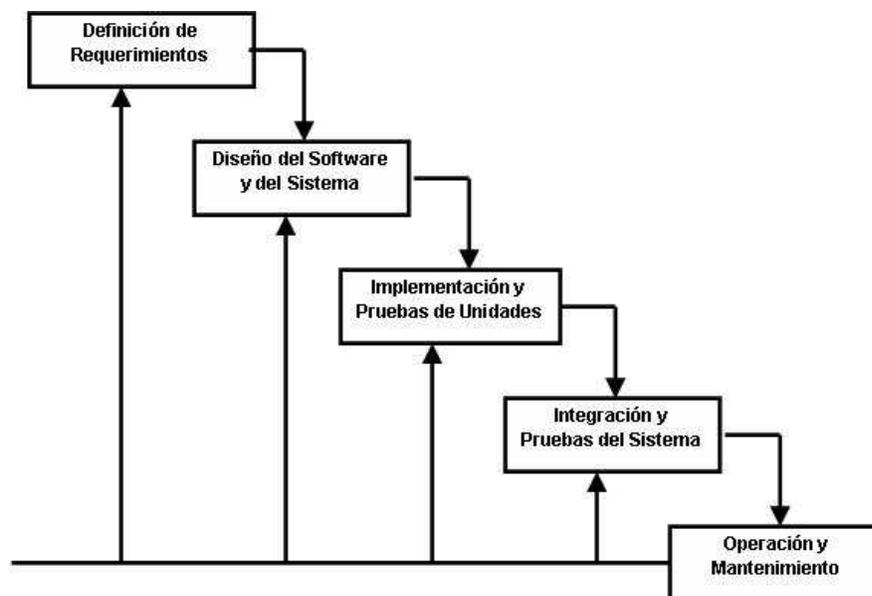
Es la etapa final del ciclo de desarrollo y representa la liberación y mantenimiento del producto terminado; también es en esta fase que se utiliza la mayor parte del tiempo de todo el ciclo de vida.

## 4.2 Métodos de desarrollo de software

### 4.2.1 Método en cascada

Es un método de desarrollo en forma secuencial, esto significa que separa distintas fases de especificación y desarrollo y para iniciar con una nueva fase debe haberse finalizado la anterior; este método tiene las desventajas siguientes: la primera, el producto se le entrega al usuario hasta el final cuando ya todo está terminado, esto es, que el usuario no observa el avance del mismo ni se involucra más que en la etapa inicial donde contribuye en la definición de requerimientos y la segunda, es que presenta dificultad para realizar cambios entre etapas.

Figura 4. Etapas del método en cascada



#### **4.2.1.1 Etapas del método en cascada**

- **Definición de requerimientos**

Es la etapa inicial del método en cascada, donde se recopila la información de los requisitos con los que debe cumplir el producto final, tales como las funciones que debe desempeñar, el rendimiento que debe ofrecer, interfaces requeridas, en general, se determina el ámbito de la información del software. Es una etapa muy importante ya que de esta etapa depende el resto y dado que es un método secuencial, si algún requerimiento no está debidamente especificado el impacto de esto se verá al finalizar el desarrollo.

- **Diseño del software y del sistema**

Esta etapa se puede definir como la traducción de los requerimientos en una representación de software; está enfocada en la definición de estructuras de datos y arquitectura apropiada al sistema a desarrollar, detalle de procedimientos y diseño de la interfaz que se presentará al usuario, todo esto es necesario para comenzar con la codificación.

- **Implementación y pruebas de unidades**

Es la traducción del diseño del sistema a un lenguaje que pueda interpretar la máquina. Esto es lo que se denomina codificación del sistema. En esta etapa también se van realizando pruebas independientes en cada módulo para verificar que lo codificado coincide con lo solicitado.

- **Integración y pruebas del sistema**

Las pruebas que se realizan en esta etapa buscan asegurar que una entrada definida produzca los resultados requeridos. Con esto se prueba la lógica interna del sistema y el funcionamiento de interfaces todo esto, de acuerdo a lo definido, diseñado y codificado en las etapas previas.

- **Operación y mantenimiento**

El sistema entregado al cliente, puede sufrir cambios ya sea por errores que se hayan encontrado, cambio en las reglas del negocio, ampliación de funcionalidades o rendimiento, cambio en hardware o sistema operativo, en fin, cualquier tipo de cambio después de completados los requerimientos planteados al inicio, se realizan en esta etapa. Las principales desventajas que se encuentran al utilizar un modelo en cascada son:

- Por parte del cliente, además de que debe ser paciente ya que será hasta el final que pueda observar una versión operativa del sistema, es muy difícil que desde el inicio pueda establecer explícitamente todos los requerimientos observando este impacto en el producto final.
- En todo proyecto siempre hay iteraciones, lo que significa que es muy poco probable que pueda desarrollarse un sistema utilizando un modelo secuencial.

#### **4.2.2 Modelo V**

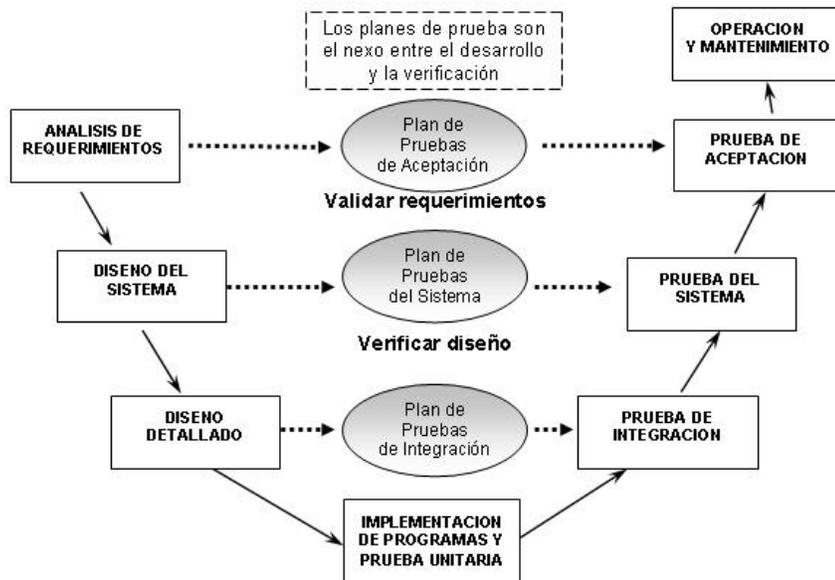
Este modelo puede definirse como una evolución del modelo en cascada ya que las primeras fases del ciclo de este modelo están representadas por las

etapas del modelo en cascada con la diferencia que se incluye una revisión en cada etapa y el resto de fases se definen para hacer pruebas e integración asociadas a cada una de las etapas iniciales.

A pesar de ser un modelo más completo y robusto, en comparación con el modelo en cascada, presenta ciertas desventajas entre las que se pueden mencionar las siguientes:

- No se contempla la posibilidad de regresar a etapas inmediatamente anteriores.
- Las pruebas de aceptación no se realizan al final de cada etapa sino después de finalizada la implementación, lo que podría conllevar la reversión completa de todo el proceso.

**Figura 5. Etapas del modelo V**



### 4.2.3 Modelo de construcción de prototipos

Un prototipo se define como: “Un modelo preliminar (representación, demostración o simulación) fácilmente ampliable o modificable de un sistema planificado previamente, incluyendo interfaz y funcionalidad de entradas y salidas”<sup>2</sup>

A través de los prototipos se puede obtener retroalimentación del usuario final, lo que reduce el riesgo de desarrollar productos que no satisfacen las necesidades del usuario aumentando la probabilidad de éxito del software, lo que lleva también a la reducción de costos. Cabe mencionar que un prototipo no presenta calidad ni robustez y para desarrollarlos se requiere de herramientas adecuadas.

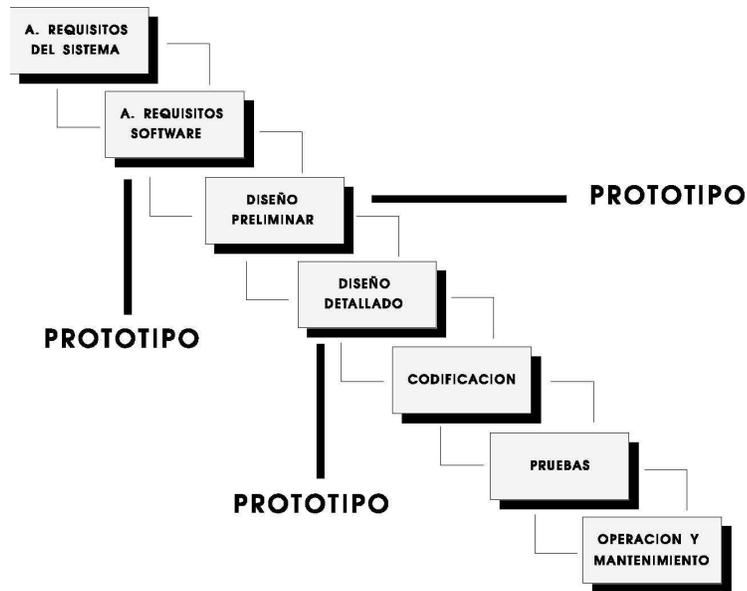
En este modelo no se modifica el flujo del ciclo de vida del software y para que sea efectivo se debe contar con un grupo de desarrollo reducido y con las herramientas y lenguajes adecuados, además el sistema a desarrollar debe permitir experimentarse y ser comparativamente barato; el desarrollo debe ser rápido con énfasis en la interfaz del usuario.

A pesar de sus ventajas, presenta riesgos tales como que el usuario puede desilusionarse al saber que lo que se le presente es solamente un prototipo y no el sistema terminado y por parte del desarrollador, puede existir el riesgo de que solamente amplíe el prototipo para construir el producto final sin tomar en cuenta la calidad y mantenimiento del software.

---

<sup>2</sup> Prototipado. <http://www.sidar.org/recur/desdi/traduc/es/visitable/maner/Prototipado.htm>

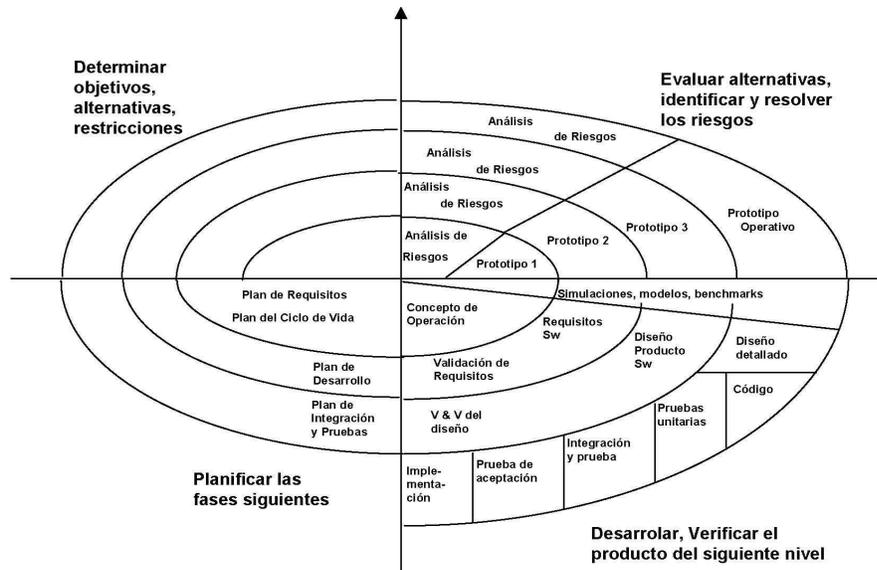
**Figura 6. Etapas del modelo de prototipos**



#### **4.2.4 Método de espiral**

Este es un modelo de desarrollo orientado a la reducción de riesgos de trabajo. Las fases de este modelo están representadas por bucles que forman una espiral. Cada bucle o iteración está conformado por un conjunto de actividades y cada actividad se define en base a un análisis de riesgo comenzando por el bucle anterior.

**Figura 7. Fases del método en espiral**



#### 4.2.4.1 Actividades del Método de Espiral

Estas actividades se definen por cada bucle que conforma la espiral de este modelo.

- **Determinar objetivos, alternativas y restricciones**

Esta actividad consiste en identificar objetivos específicos para cada fase del proyecto, además de establecer los productos a obtener: requerimientos, especificación, manual de usuario, restricciones. También se identifican riesgos del proyecto y estrategias para evitarlos.

- **Evaluar alternativas, identificar y resolver riesgos**

Esta actividad consiste en identificar todos los riesgos potenciales y plantear y determinar estrategias alternativas para reducirlos o resolverlos

- **Desarrollar, verificar y validar**

Esta actividad está conformada por el desarrollo y las pruebas de cada fase; dependiendo del análisis de riesgos realizado previamente se establece el modelo más apropiado para realizar el desarrollo y cumplir con los objetivos establecidos al inicio de la fase.

- **Planificación**

Se hace una revisión de todo lo que se ha realizado en la fase, se evalúa el avance y el cumplimiento de objetivos trazados y dependiendo del resultado obtenido, se decide si se continúa con las fases siguientes y se planifica la siguiente actividad.

#### **4.2.5 *Rational Modified Process (RUP)***

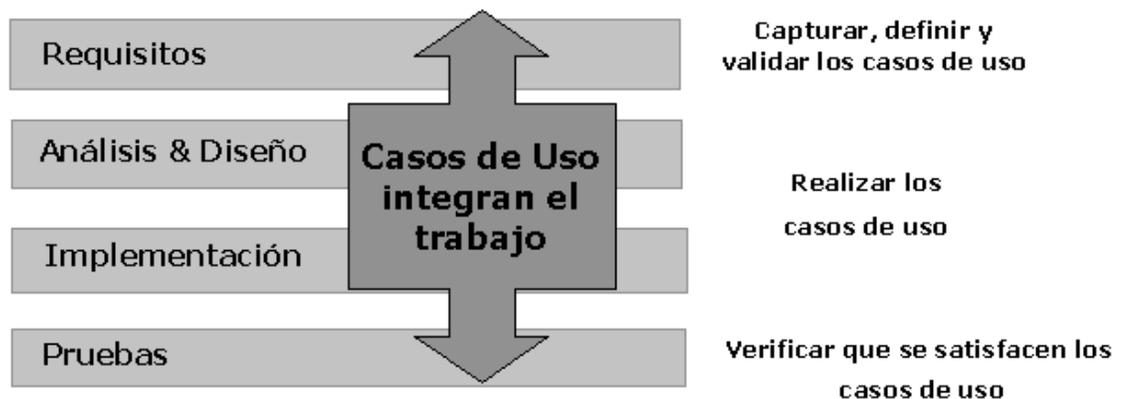
##### **4.2.5.1 Características esenciales**

El proceso de software propuesto por RUP tiene tres características esenciales: está dirigido por los casos de uso, está centrado en la arquitectura y es iterativo e incremental.

#### 4.2.5.1.1 Proceso dirigido por casos de uso

En RUP los casos de uso no sólo representan los requerimientos funcionales del sistema, sino que también guían su diseño, implementación y pruebas. A su vez proporcionan un hilo conductor que permite establecer la trazabilidad entre artefactos generados en las distintas actividades del proceso de desarrollo.

**Figura 8. Integración del trabajo**



#### 4.2.5.1.2 Proceso centrado en la arquitectura

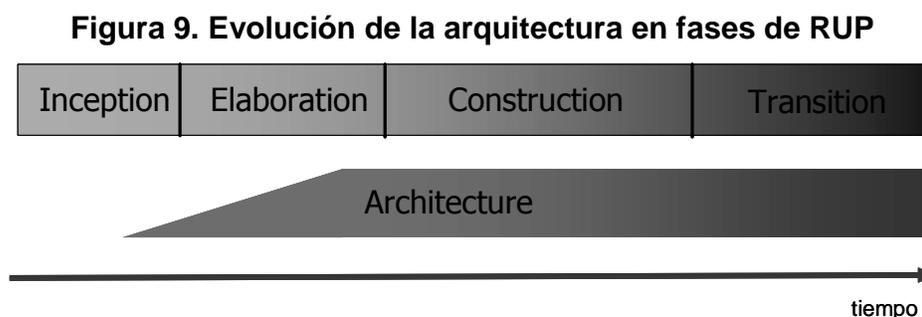
La arquitectura de un sistema es la organización o estructura de sus partes más relevantes, lo que permite tener una visión común entre todos los involucrados (desarrolladores y usuarios) y una perspectiva clara del sistema completo, necesaria para controlar el desarrollo.<sup>3</sup>

<sup>3</sup> Rational Unified Process. <http://ivanex.wikidot.com/metodologia>

La definición de la arquitectura se hace con base al sistema operativo, protocolos, sistema manejador de bases de datos, entre otras características, considerando también aspectos que constituyen la calidad del sistema, tales como la extensibilidad, la reutilización y el rendimiento ya que es la arquitectura parte esencial para determinar como y el orden en el que debe construirse un sistema.

La forma de cada producto es proporcionada por la arquitectura y la función del mismo es la que se refleja en los casos de uso, es por ello que tanto la arquitectura como los casos de uso evolucionan paralelamente en el proceso de desarrollo de software.

En las fases iniciales del proyecto, se consolida la arquitectura a través de *baselines* y se modifica dependiendo de las necesidades del proyecto, por lo que la arquitectura final es más robusta.



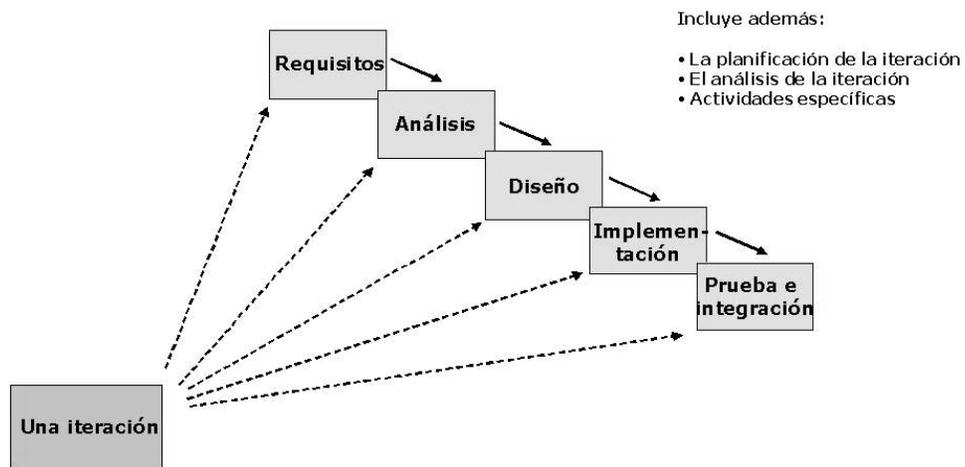
#### 4.2.5.1.3 Proceso iterativo e incremental

El proceso iterativo e incremental consta de una serie de iteraciones donde cada iteración aborda una parte de la funcionalidad total, pasando por todos los flujos de trabajo relevantes y refinando la arquitectura. Este proceso es una estrategia que utiliza RUP para dividir el trabajo en partes más

pequeñas, estableciendo un equilibrio entre arquitectura y casos de uso en todo el proceso de desarrollo. En cada iteración se obtiene un incremento en el producto final.

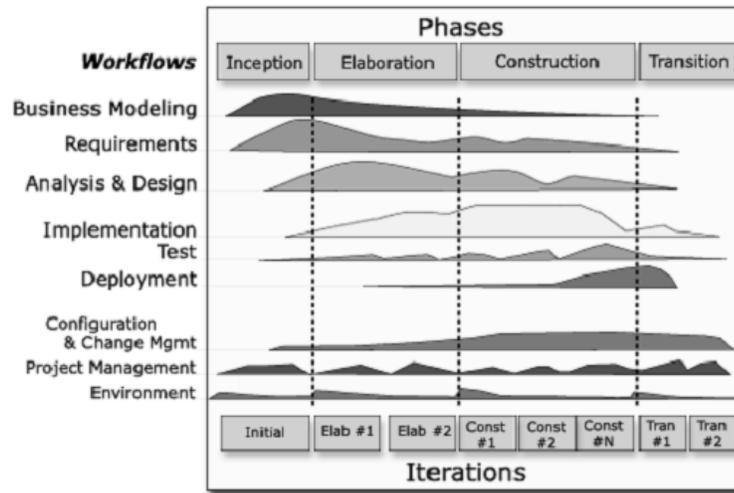
Una iteración puede estar constituida por un ciclo del modelo en cascada y al finalizar, los resultados obtenidos se integran con los resultados de iteraciones anteriores.

**Figura 10. Iteración en RUP**



El proceso de RUP está dividido en cuatro fases dentro de las que se realiza un número determinado de iteraciones que dependen de cada proyecto. En la gráfica que se presenta a continuación, se puede observar el esfuerzo necesario en cada fase.

**Figura 11. Esfuerzo en cada fase de RUP**



Las iteraciones iniciales en las fases de inicio y elaboración, están enfocadas en establecer aspectos como el ámbito del proyecto y una línea base de la arquitectura, además de eliminar riesgos críticos y comprender la tecnología y el problema a resolver con el sistema.

En la fase de inicio, las iteraciones ponen mayor énfasis en el modelado del negocio y requerimientos; en la fase de elaboración, las iteraciones están orientadas al desarrollo de la línea base de la arquitectura; en la fase de construcción se desarrolla el producto a través de iteraciones y en la fase de transición se busca garantizar que el producto esté preparado para su entrega.

## **5. SITUACIÓN ACTUAL DE LOS SISTEMAS DE INFORMACIÓN EN GUATEMALA**

### **5.1 Introducción**

Basado en lo que se ha expuesto anteriormente, el trabajo que se entrega en este capítulo presenta los resultados obtenidos a través de una encuesta cuyo objetivo fue recopilar información que permita determinar la situación de los sistemas de información en Guatemala, específicamente en los siguientes puntos: Plataformas, metodologías de desarrollo y arquitecturas; así como los niveles, estrategias y herramientas para la integración de sistemas de información. Este estudio fue realizado por un período de tres meses comprendidos de agosto a noviembre del año 2007.

Se contó con la participación de 52 personas, quienes son profesionales de la informática e interactúan directamente con los sistemas de información de diversas organizaciones en el país.

La encuesta está dividida en dos partes, la primera serie de preguntas está enfocada a conocer la población que respondió la encuesta y a su vez a determinar si es una organización que cuenta con sistemas heterogéneos en los que aplique la integración de sistemas. La segunda parte, está compuesta por preguntas relacionadas al tema de integración de sistemas de información, donde se busca conocer plataformas de desarrollo, herramientas y tecnologías para integración de sistemas, metodologías de desarrollo y herramientas para administración de información, entre otros aspectos.

## 5.2 Detalle de la encuesta y resultados

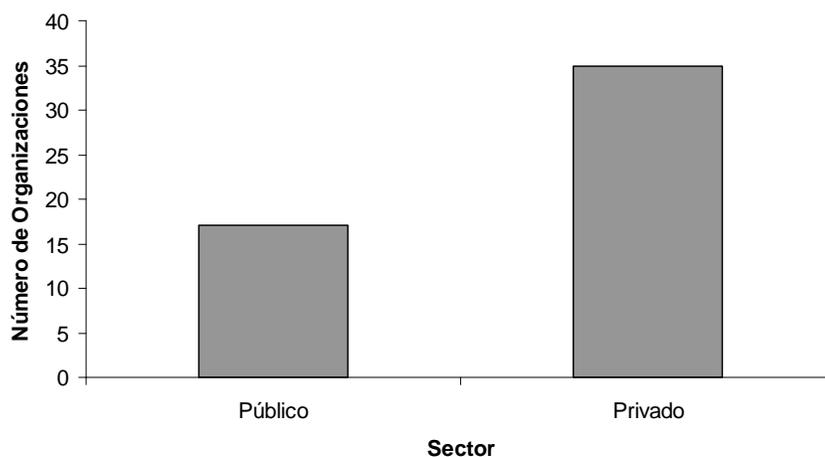
### 5.2.1 Sector de la organización

La finalidad de esta pregunta es conocer el origen de las aportaciones del capital de la organización así como a quienes van dirigidas las actividades que realiza la empresa para la que trabaja el encuestado, para poder determinar esto, las opciones que se colocaron a los encuestados fueron:

- a. Público
- b. Privado.

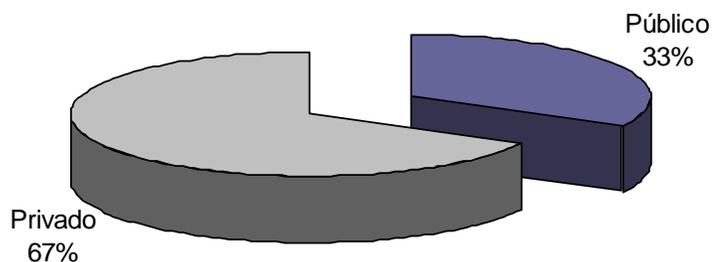
Los resultados de esta pregunta, se pueden observar en las siguientes gráficas.

**Figura 12. Organizaciones por sector**



Los resultados indican que el 33% de los encuestados trabajan para organizaciones públicas y el 67% para el sector privado.

**Figura 13. Organizaciones por sector (en proporciones)**



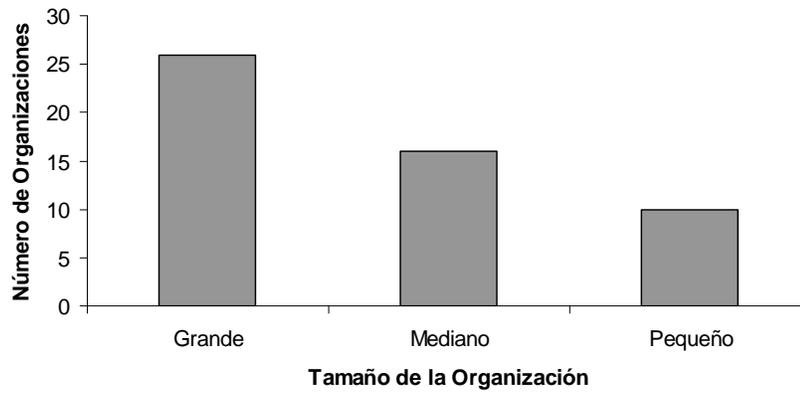
### **5.2.2 Tamaño de la organización**

La finalidad de esta pregunta es conocer el tamaño de la organización para la que labora el encuestado o le pertenece al encuestado, esto basado en el nivel de producción, recursos económicos y tecnológicos, entre otros. Las opciones presentadas al encuestado fueron:

- a. Pequeño
- b. Mediano
- c. Grande

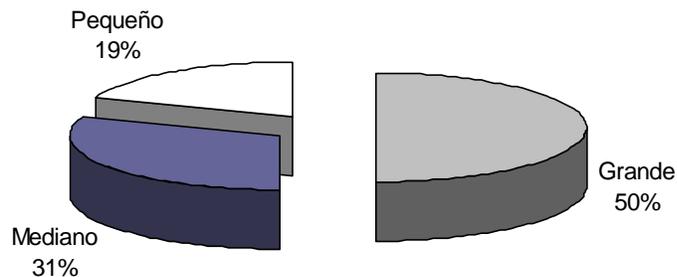
Los resultados obtenidos fueron los siguientes:

**Figura 14. Organizaciones por tamaño**



Lo que nos indica que del total de organizaciones, el 50% se clasifica como empresa de grande, el 31% como empresa mediana y el 19% es una empresa pequeña.

**Figura 15. Organizaciones por tamaño (en proporciones)**



### 5.2.3 Rol de la organización

El objetivo de esta pregunta es conocer la actividad económica que realiza la empresa para la que trabaja o que pertenece al encuestado; para ello

se realizó una clasificación dependiendo de las distintas áreas donde se puede aplicar la integración de sistemas; al encuestado se le presentaron las siguientes opciones:

- a. Desarrollo de SW
- b. Banca y finanzas
- c. Comercial
- d. Distribución
- e. Educación
- f. Otros especifique\_\_\_\_\_

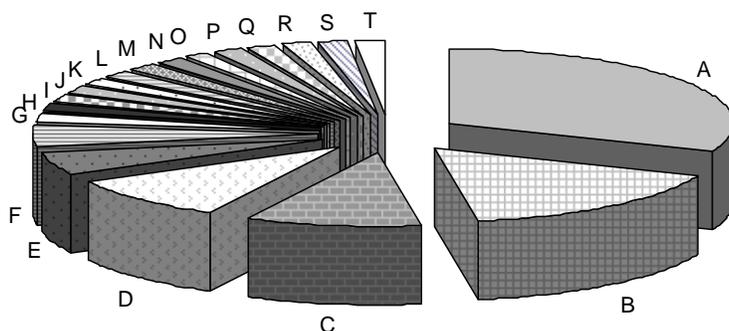
Los resultados para esta pregunta se presentan en la siguiente tabla y su respectiva gráfica.

**Tabla I. Roles de las organizaciones**

<b>Rol</b>	<b>Descripción</b>	<b>Cantidad</b>	<b>Porcentaje</b>
A	Desarrollo de SW	19	31.15
B	Educación	10	16.39
C	Banca y finanzas	6	9.84
D	Comercio	6	9.84
E	Distribución	3	4.92
F	Recaudación de impuestos	3	4.92
G	Rediseño de procesos de negocio	1	1.64
H	Artes gráficas	1	1.64
I	Analistas	1	1.64
J	Seguridad social	1	1.64
K	Telecomunicaciones	1	1.64
L	Judicial	1	1.64
M	Apoyo transversal de TICs a sistemas de integración financiera	1	1.64

N	Registro de posesión de tierra	1	1.64
O	Administración de proyectos	1	1.64
P	Generación de energía eléctrica	1	1.64
Q	Consultoría en soluciones informáticas	1	1.64
R	Infraestructura y tecnología	1	1.64
S	Salud	1	1.64
T	Alimentación	1	1.64

**Figura 16. Proporciones de los roles de las organizaciones**



De un total de 52 encuestas válidas, el 32.69% fue contestado por profesionales que laboran para el sector público del país y el resto, el 67.31%, por personal que trabaja en forma independiente o labora para organizaciones privadas.

Las actividades económicas que realizan las organizaciones encuestadas, se detallan en la siguiente tabla, agrupados por sector empresarial y los datos se muestran en porcentajes.

**Tabla II. Actividades de la población encuestada (en porcentajes)**

Actividad	Sector Público	Sector Privado
Desarrollo de SW	8.20%	22.95%
Educación	6.56%	9.84%
Banca y finanzas	0%	9.84%
Comercio	0%	9.84%
Distribución	0%	4.92%
Recaudación de impuestos	4.92%	0%
Rediseño de procesos de negocio	1.64%	0%
Artes gráficas	0.00%	1.64%
Analistas	1.64%	0%
Seguridad social	1.64%	0%
Telecomunicaciones	1.64%	0%
Justicia	1.64%	0%
Apoyo transversal de TICs a sistemas de integración financiera	1.64%	0%
Registro de posesión de tierra	1.64%	0%
Administración de proyectos	0%	1.64%
Generación de energía eléctrica	0%	1.64%
Consultoría en soluciones informáticas	0%	1.64%
Infraestructura y tecnología	0%	1.64%
Salud	0%	1.64%
Alimentación	0%	1.64%

De estas organizaciones, el 50% está representado por empresas grandes; el 30.77% por empresas medianas y el 19.23% por empresas pequeñas. Además, cabe destacar el ámbito que abarcan estas empresas, ya que el 75% de las organizaciones son nacionales; el 21.16% son multinacionales y el 3.85% son transnacionales.

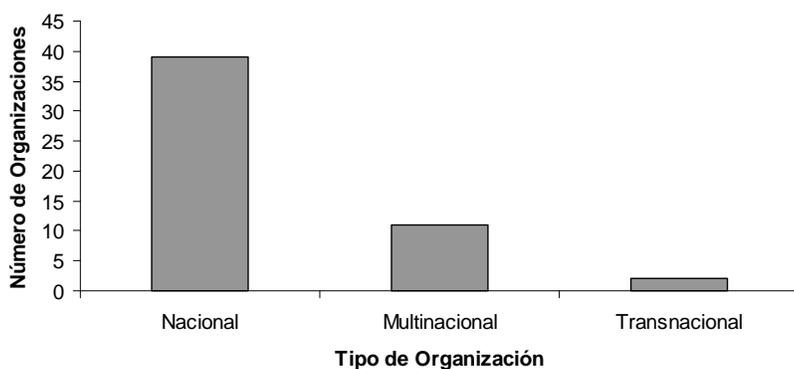
#### 5.2.4 Tipo de organización

La finalidad de esta pregunta es determinar el ámbito geográfico en el que la organización realiza su actividad, para ello, se presentaron las siguientes opciones:

- a. Nacional
- b. Multinacional
- c. Transnacional
- d. Otro especifique \_\_\_\_\_

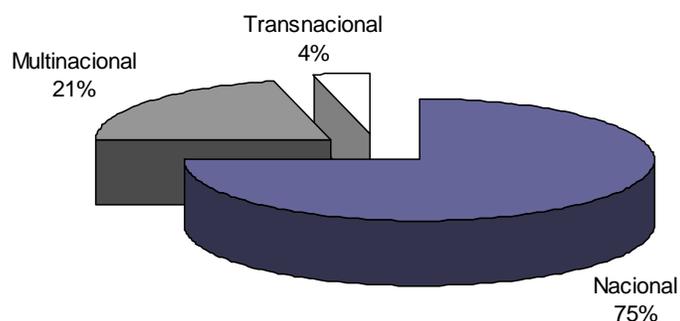
Los resultados de esta pregunta se puede observar en las siguientes gráficas:

**Figura 17. Tipo de organización**



Lo que indica que el 75% de organizaciones es nacional, el 21% es multinacional y el 4% es transnacional.

**Figura 18. Tipo de organización (en proporciones)**



### **5.2.5 Sistemas heterogéneos**

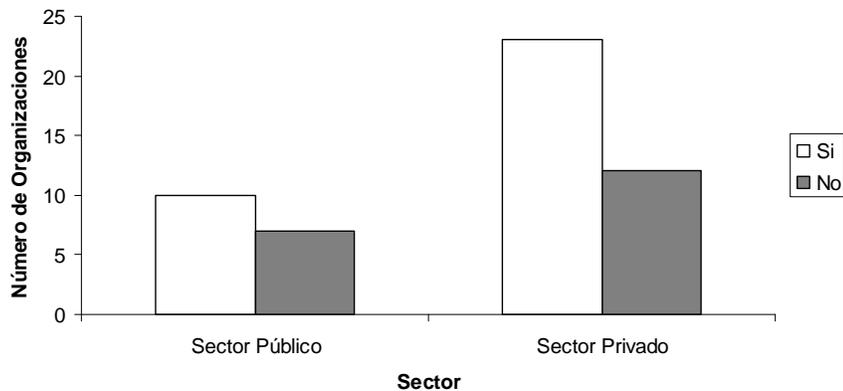
Para poder continuar con el estudio, es necesario determinar si las organizaciones encuestadas cuentan con sistemas heterogéneos, esto debido a que es el punto de partida para poder consultar acerca de cómo realizan la interacción entre sistemas, que es la finalidad principal del estudio presentado.

La pregunta presentada fue: ¿Posee en su organización sistemas heterogéneos? Las respuestas que el encuestado podía seleccionar fueron:

- a. Sí
- b. No

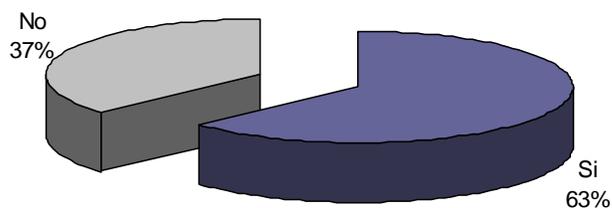
Los resultados obtenidos se pueden observar en las siguientes gráficas:

**Figura 19. Sistemas heterogéneos por sector**



De acuerdo a la gráfica anterior, se puede determinar que del total de empresas encuestadas, el 63.46% tienen sistemas heterogéneos para el desempeño de sus actividades, correspondiendo un 19.23% al sector público y un 44.23% al sector privado. A continuación se muestran los resultados en proporción por cada respuesta.

**Figura 20. Proporción de organizaciones que poseen sistemas heterogéneos**



En la encuesta, se indicó que si la respuesta a esta pregunta era “Sí”, que por favor continuara con las preguntas siguientes.

### **5.2.6 ¿Están integrando sus sistemas las empresas?**

La integración de sistemas es una necesidad que las empresas de Guatemala han tenido que satisfacer para poder automatizar cada uno de sus procesos, logrando con ello, mayor exactitud y rapidez en la realización de sus operaciones.

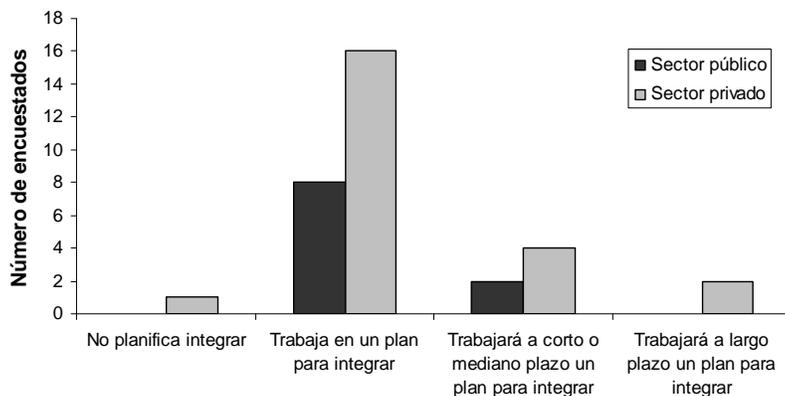
Para obtener los resultados a esta interrogante, se solicitó al encuestado, después de haber indicado que sí poseía sistemas heterogéneos la organización para la que trabaja, que seleccionara la opción más adecuada a su situación, en base al siguiente enunciado:

Para la integración de estos sistemas heterogéneos

- a. Ya está trabajando en un plan para integrarlos
- b. Trabaja a un corto o mediano plazo un plan para integrarlos
- c. Trabaja a un largo plazo un plan para integrarlos
- d. No está planificado integrarlos

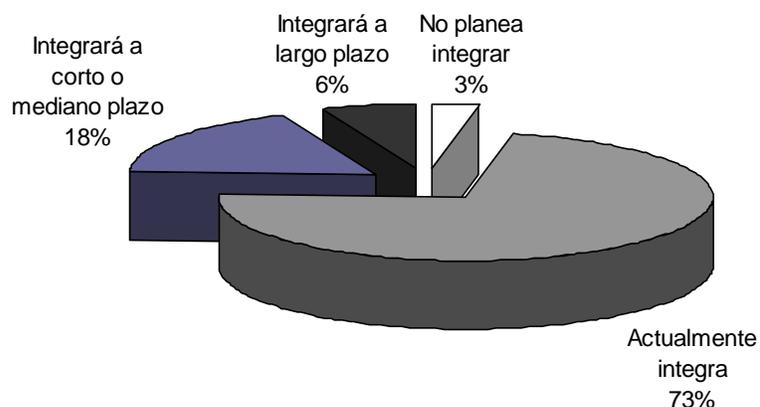
Los resultados obtenidos se pueden observar en las siguientes gráficas.

**Figura 21. Integración de sistemas en Guatemala**



Tal como se puede observar, el porcentaje de empresas que no tiene planificado integrar sus sistemas es bastante bajo (3.03%). Una de las causas por las que una empresa no tenga planificado integrar sus sistemas, puede ser porque que sus sistemas ya están optimizados y la integración de los mismos ya esté definida y administrada. Por otro lado, la mayoría de empresas ya está integrando sus sistemas (72.73%) y otras, ya tienen la visión de la integración que desean realizar, ya sea a corto o mediano plazo (18.18%) o a largo plazo (6.06%).

**Figura 22. Integración de sistemas heterogéneos (proporciones)**



### 5.2.7 Plataformas de desarrollo

Es necesario conocer las plataformas sobre las que son desarrollados los sistemas que participan en la integración, como parte del estudio, es interesante saber las distintas plataformas que actualmente están incluyendo las organizaciones en sus sistemas y con las que se ha logrado integración de sus sistemas.

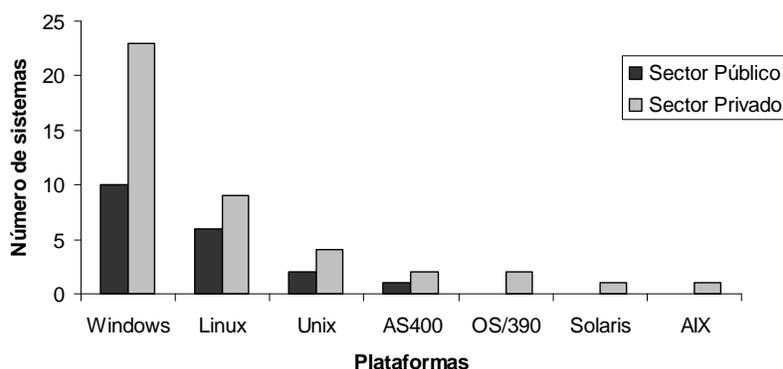
Para poder obtener estos datos, se planteó la pregunta que se presenta a continuación y también las respuestas que podía seleccionar el encuestado:

¿En qué plataforma se encuentran los sistemas que integra o integrará?

- a. Windows
- b. Linux
- c. Unix
- d. Otros especifique \_\_\_\_\_

Los resultados de esta pregunta se presentan en la siguiente gráfica, donde se detalla la combinación de plataformas que utilizan las organizaciones en sus sistemas.

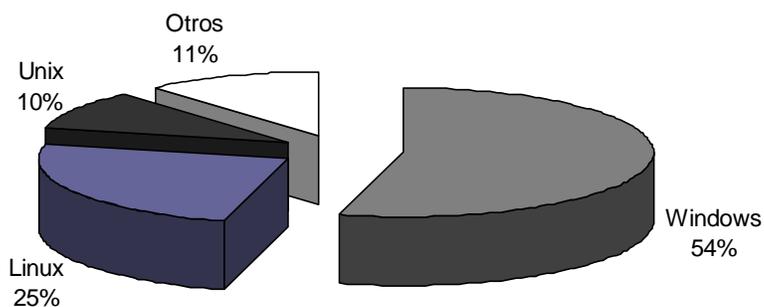
**Figura 23. Plataformas de desarrollo utilizadas en los sistemas de información en Guatemala**



Como se puede observar en la gráfica anterior, la plataforma más utilizada, tanto en el sector público como privado, es la plataforma Windows, lo que podría interpretarse de la siguiente forma: En cada una de las organizaciones existe al menos un sistema que funciona sobre esta plataforma.

En la gráfica siguiente se puede observar la proporción de cada plataforma, entre las que destaca Windows con el 54.09%, Linux con el 24.59%, Unix con el 9.83% y el resto de plataformas con el 11.47%

**Figura 24. Plataformas de desarrollo utilizadas (en proporciones)**



### **5.2.8 Herramientas para administración de información**

Actualmente, todas las empresas incluidas en este estudio, utilizan en sus distintos sistemas, al menos un Sistema Administrador de Bases de Datos (DBMS) y las Bases de Datos que utilizan son relacionales. Sin embargo, no dejan de apoyarse en otras herramientas como lo es el Método de Acceso de Almacenamiento Virtual (VSAM).

Las herramientas que se utilicen pueden variar, dependiendo de las necesidades de cada organización y la cantidad de información que tenga que administrar.

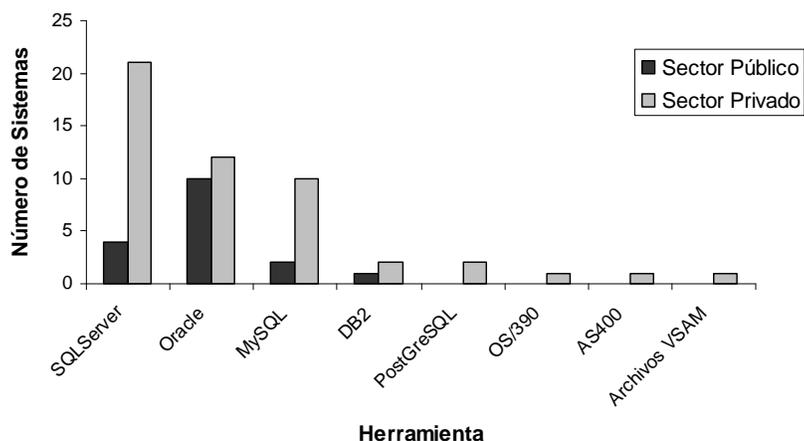
Para poder determinar estos datos, a los encuestados se les presentó la siguiente pregunta con sus respectivas opciones:

¿Qué manejadores de Bases de Datos utilizan los sistemas que integra o integrará?

- a. SQLServer
- b. Oracle
- c. MySQL
- d. Otros especifique \_\_\_\_\_

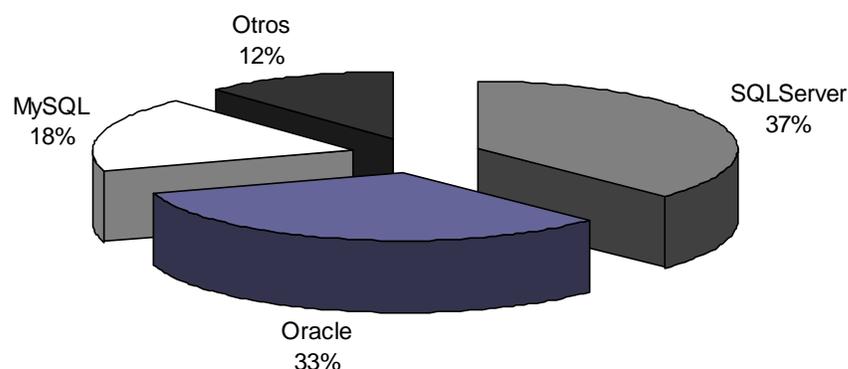
Los resultados de esta pregunta, se pueden observar en las siguientes gráficas:

**Figura 25. Herramientas para administración de la información utilizadas en los sistemas de información en Guatemala**



Las proporciones de acuerdo a cada Sistema Manejador de Bases de Datos, se presentan en la siguiente gráfica, estando distribuidos los resultados de la siguiente manera: SQLServer 37.31%, Oracle 32.84%, MySQL 17.91% y el resto con 11.94%

**Figura 26. Herramientas para administrar información (en proporciones)**



### **5.2.9 ¿Cómo integrarán sus sistemas las empresas?**

Esta pregunta está enfocada a conocer la forma en la que las empresas integraran sus sistemas, es decir, si la integración de los sistemas que utilizan, serán desarrollados internamente, pagaran a terceros para integrarlos o utilizaran una combinación de las modalidades mencionadas previamente.

El enunciado y las posibles respuestas presentadas al encuestado para poder obtener este dato fue:

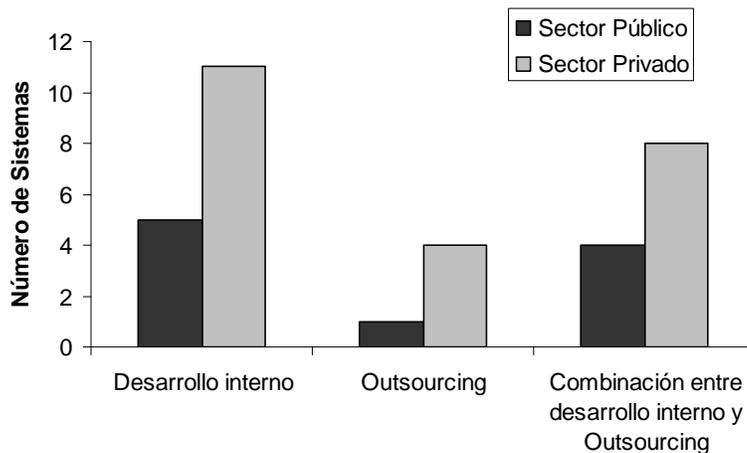
La integración la realiza o realizará de la siguiente forma

- a. Desarrollo interno
- b. Outsourcing
- c. Combinación entre desarrollo interno y *Outsourcing*
- d. Otro especifique \_\_\_\_\_

La forma en la que las empresas desarrollan sus sistemas puede depender mucho de la actividad que realizan, principalmente por el nivel de confidencialidad que debe manejarse con la información que administra.

Además de considerar otros aspectos como el costo que conlleva este desarrollo, encontrando un balance al desarrollar internamente, hacerlo a través de terceros o combinando ambas opciones. Las tendencias en el desarrollo de sistemas, se muestran en la siguiente gráfica:

**Figura 27. Modalidad para desarrollo de sistemas en Guatemala**

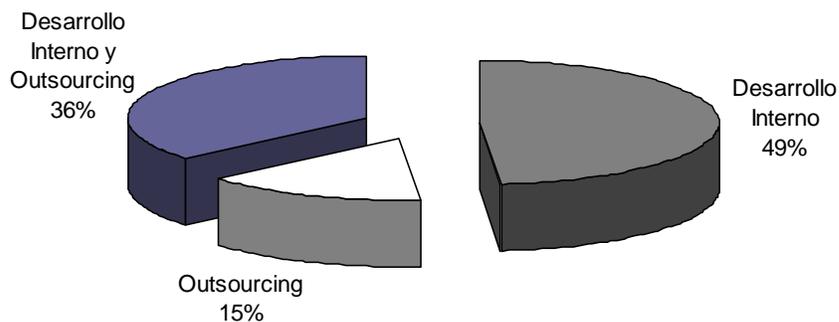


Esto da como resultado que el 48.48% de las empresas encuestadas tienen sus propios desarrolladores. Esta forma de desarrollo la utiliza el 35.29% de las empresas grandes, el 41.18% de empresas medianas y el 23.53% de empresas pequeñas.

Muchas de las empresas tienen como segunda opción la combinación de desarrollo interno y desarrollo por parte de terceros, muestra de ellos es que el 36.36% de las organizaciones desarrolla sus sistemas de esta forma.

Y finalmente, también hay empresas que todo el desarrollo de sus sistemas es a través de *outsourcing* (desarrollo por parte de terceros), esto está representado por el 15.15% de las empresas.

**Figura 28. Modalidad para integrar sistemas de información (en proporciones)**



#### **5.2.10 Metodologías para integrar sistemas de información**

La metodología que se escoja para la integración de un sistema, dependerá mucho de la funcionalidad del sistema, de la cantidad de personas que participarán en el desarrollo y del tiempo que se tiene disponible para el mismo.

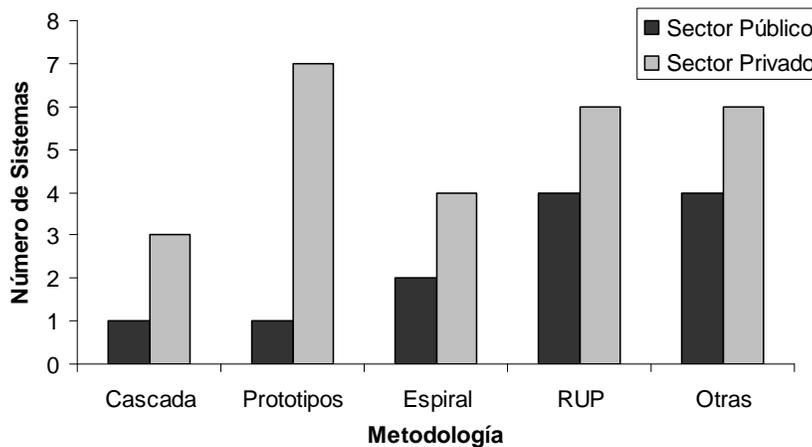
Esta pregunta está orientada a saber cual es la metodología más utilizada por empresas guatemaltecas para desarrollar *software* y como es que integran sus sistemas; así como también poder conocer hacia donde se están orientando las empresas en cuanto se refiere a las metodologías de desarrollo; para ello, al encuestado se le solicitó responder a la siguiente pregunta:

¿Qué metodología utiliza o utilizará para realizar esta integración?

- a. Cascada
- b. Modelo V
- c. Prototipos
- d. Espiral
- e. RUP
- f. Otros Especifique \_\_\_\_\_

Los resultados de esta pregunta, se pueden observar en las siguientes gráficas:

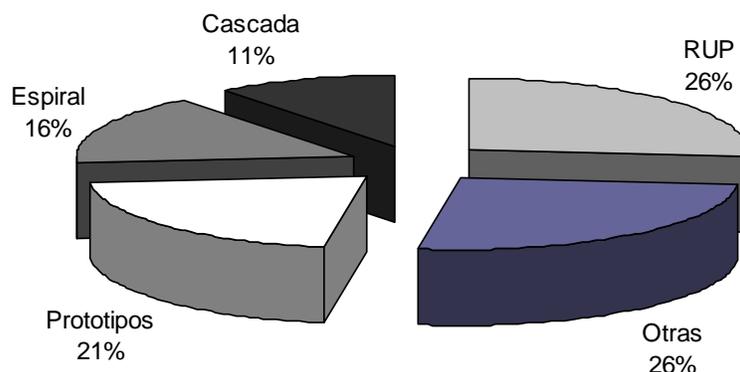
**Figura 29. Metodologías de desarrollo utilizadas en sistemas de información en Guatemala**



De estos resultados, las proporciones de cada una de las metodologías son el 26.32% para RUP, el 26.32% para otras metodologías (no especificadas), el 21.05% para la metodología de Prototipos, el 15.79% para la metodología en Espiral y el 10.53% para la metodología en Cascada.

También se ha podido determinar que al menos el 57.58% de las empresas emplean una misma metodología de desarrollo para todos sus sistemas, el 12.12% utiliza más de una metodología y el 30.3% emplea sus propias metodologías ya sea estableciendo una nueva forma de desarrollo o formulando híbridos de las metodologías estándar.

**Figura 30. Metodologías de desarrollo (en proporciones)**



### 5.2.11 Niveles de integración de los sistemas de información

Para poder determinar el nivel de integración que tenía cada una de las empresas incluidas en este estudio, se tomó como base las siguientes descripciones para cada uno de los niveles:

- Las tecnologías utilizadas se comunican a través de interfaces intermedias con una baja interrelación entre los sistemas (Nivel 1)
- Utiliza tecnología *Middleware* que consolida las reglas del negocio y las transacciones entre aplicaciones; existe una estructura central que controla el intercambio de información (Nivel 2)

- Maneja un flujo de información entre aplicaciones. Toma en cuenta las reglas del negocio y el flujo de procesos (Nivel 3)
- Utiliza tecnologías EAI que unen directamente a clientes y proveedores para operaciones internas (Nivel 4)

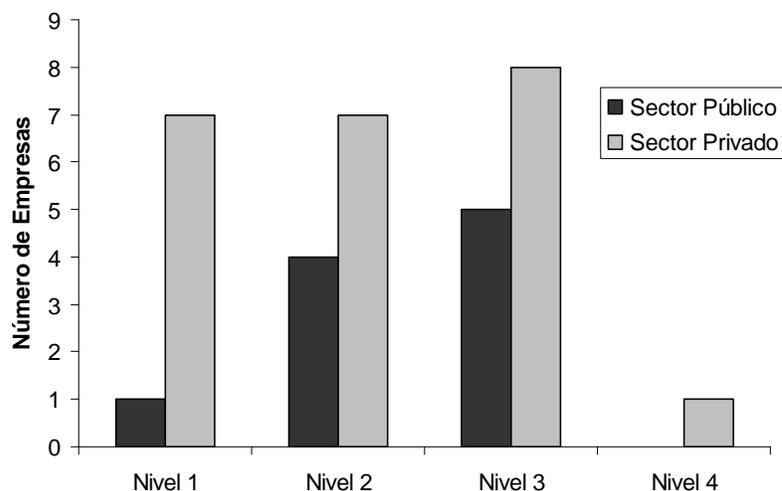
El objetivo de este punto es conocer los niveles de integración de sistemas de información que manejan las empresas guatemaltecas y poder tener un punto de referencia que pueda indicarnos la situación actual en nuestro país, para esto, el enunciado y las posibles respuestas presentadas fueron:

Seleccione el nivel de integración que mejor describe el proceso que actualmente se realiza en su empresa.

- a. Utiliza tecnologías EAI que unen directamente a clientes y proveedores para operaciones internas.
- b. Las tecnologías utilizadas se comunican a través de interfaces intermedias con una baja interrelación entre los sistemas.
- c. Utiliza tecnología *Middleware* que consolida las reglas del negocio y las transacciones entre aplicaciones; existe una estructura central que controla el intercambio de información.
- d. Maneja un flujo de información entre aplicaciones. Toma en cuenta las reglas del negocio y el flujo de procesos.

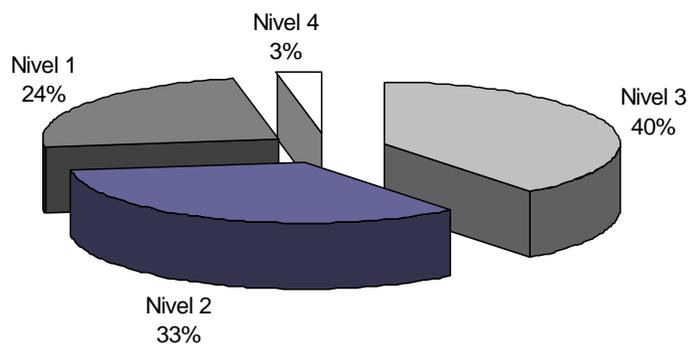
Los resultados a esta pregunta se presentan en las siguientes gráficas:

**Figura 31. Niveles de integración de sistemas en Guatemala**



Como se puede observar en la siguiente gráfica, la mayoría de las empresas que integran sistemas, están en los niveles intermedios (2 y 3), constituyendo el 72.73% de las empresas; tan solo un 3.03% se encuentra en un nivel óptimo y el 24.24% está en un nivel inicial (Nivel 1).

**Figura 32. Niveles de integración de sistemas en Guatemala (en proporciones)**



### 5.2.12 Tecnologías utilizadas para la integración de sistemas de información

En este punto se busca conocer las distintas tecnologías utilizadas por empresas guatemaltecas para integrar sistemas de información, para ello se solicitó al encuestado contestar las siguientes preguntas:

¿Qué tecnología(s) de integración utiliza actualmente?

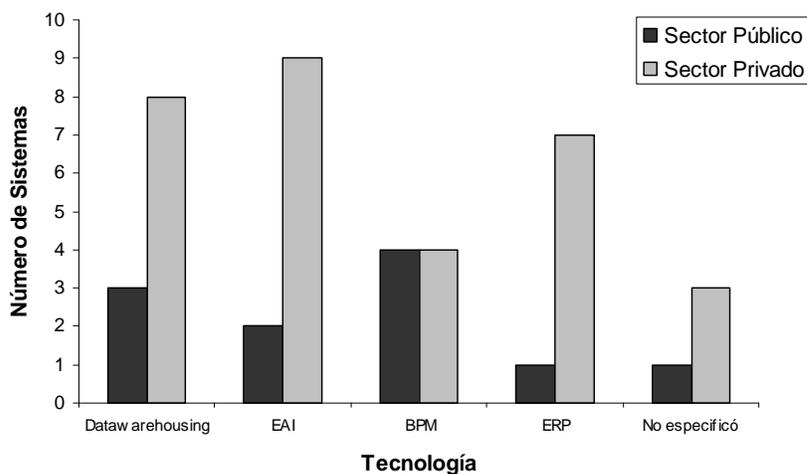
- a. Datawarehousing
- b. ERP (Enterprise Resource Planning)
- c. EAI (Enterprise Application Integration)
- d. BPM (Business Process Management)
- e. Otras especifique: \_\_\_\_\_

Si utiliza la tecnología EAI, seleccione el tipo de *Middleware*

- a. Asynchronous Event/Message Transport
- b. Transformation Engines
- c. Integration Brokers
- d. Business Process Management Framework

Los resultados de esta pregunta, se presentan en las siguientes gráficas:

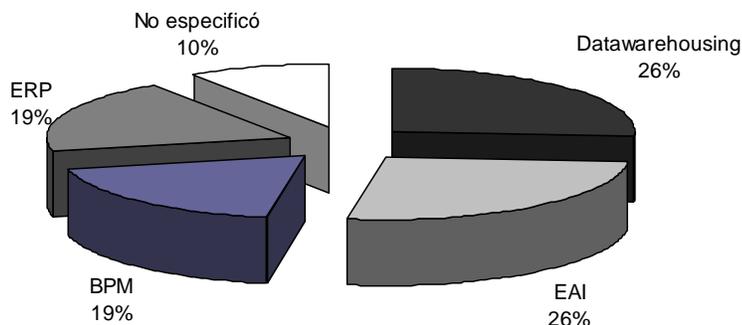
**Figura 33. Tecnologías utilizadas para la integración de sistemas en Guatemala**



Entre las tecnologías más utilizadas por la mayoría de las empresas están: *Datawarehousing* (26.19%), EAI (26.19%), ERP (9.09%), BPM (19.04%) y un 9.52% no especificó la tecnología para integrar sus sistemas.

De todas las organizaciones que integran sus sistemas utilizando una EAI, el 30% utiliza el *middleware Asynchronous Event/Message Transport*, el 20%, el *middleware Business Process Management Framework* y el 50% restante utiliza el *middleware Integration Brokers*.

**Figura 34. Tecnologías (proporciones)**



### 5.2.13 Arquitectura de los sistemas integrados

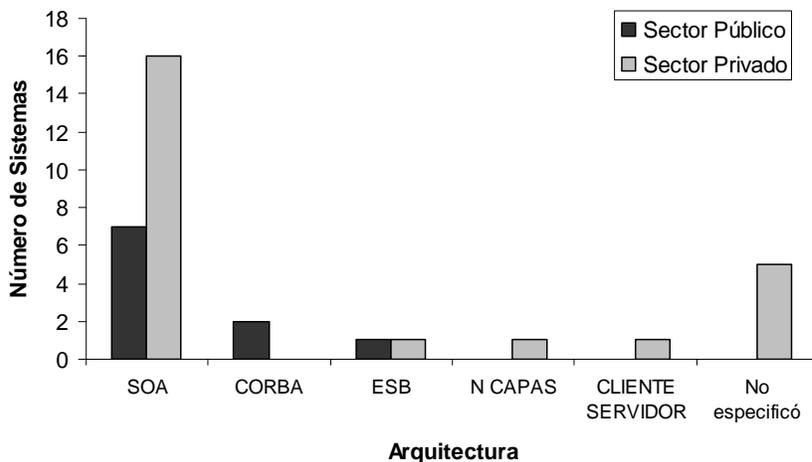
En este punto, el objetivo es conocer las distintas arquitecturas utilizadas en los sistemas involucrados en integraciones, es decir, conocer las arquitecturas entre las que ha podido establecer una interacción. Al encuestado se le presentó la siguiente pregunta y las posibles respuestas:

¿Sobre qué arquitectura funcionan los sistemas integrados?

- a. SOA (Service Oriented Architecture)
- b. ESBs (Enterprise Service Buses)
- c. CORBA
- d. Otros especifique: \_\_\_\_\_

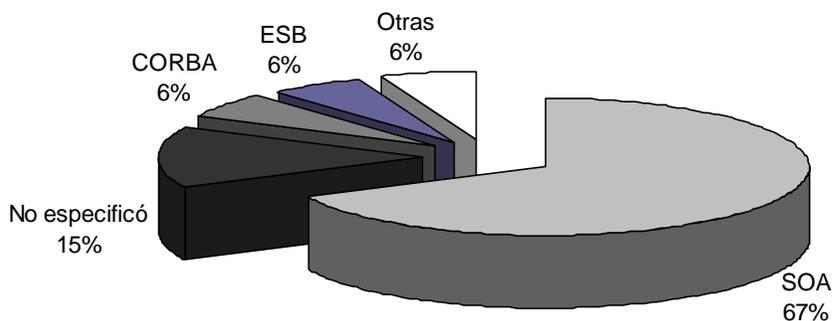
Los resultados obtenidos para esta pregunta se presentan en las siguientes gráficas:

**Figura 35. Arquitectura de sistemas integrados**



Tal como se menciona en los capítulos anteriores, entre las arquitecturas más utilizadas y que facilitan la integración de sistemas, está SOA y para este estudio, el resultado de los sistemas cuya integración funciona sobre esta arquitectura es del 66.67%; el 5.88% funciona sobre CORBA; el 5.88% sobre ESB, el 5.88% utiliza otras plataformas y el 14.71% no se especificó la arquitectura de los sistemas.

**Figura 36. Arquitectura de sistemas integrados (proporciones)**



#### 5.2.14 Herramientas más utilizadas para la integración de sistemas

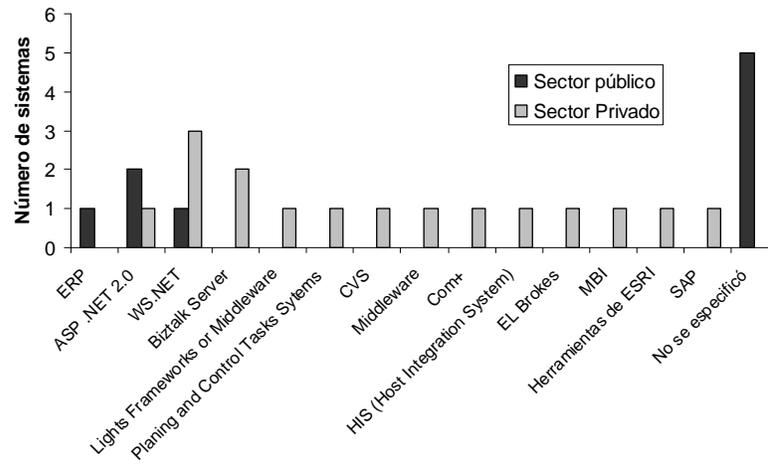
Y para finalizar la encuesta, se le solicitó al encuestado que indicara las herramientas que utilizan en la organización para la que trabaja, para llevar a cabo la integración de los sistemas de información de la misma.

El objetivo de este punto es conocer las herramientas más utilizadas en empresas guatemaltecas y de la misma manera, determinar la preferencia en tecnologías.

Entre las herramientas más utilizadas por las empresas encuestadas, se pueden mencionar:

- ERP
- ASP .NET 2.0
- Implementación de Servicios Web XML (WS.NET)
- Biztalk Server
- Light Frameworks
- Planning and Control Tasks Systems
- CVS
- *Middleware*
- Com+
- HIS (Host Integration System) permite la comunicacion de programas de COBOL (OS/390) y los traduce a una libreria TLB (que es un COM que se registra en el sistema operativo y puede ser interpretado por .NET)
- MBI
- Herramientas de ESRI

**Figura 37. Herramientas para la integración de sistemas**





## 6. TENDENCIAS Y EXPECTATIVAS EN LA INTEGRACIÓN DE SISTEMAS DE INFORMACIÓN EN GUATEMALA

### 6.1 Tendencias

Las tendencias que se presentan, de acuerdo a los resultados anteriores son:

- No importando a qué sector pertenezca una organización, las empresas de Guatemala ya están aplicando una metodología para integrar sus sistemas de información, lo que permite interconectar procesos optimizados con procesos nuevos, utilizando distintas herramientas.
- Las plataformas sobre las que funcionan los sistemas informáticos de las organizaciones en Guatemala, varían de acuerdo al tipo de actividad productiva que desempeñe la organización, ya que existen sistemas que administran su propia plataforma, tal es el caso de sistemas que utilizan organizaciones gubernamentales y el área de banca y finanzas. La tendencia que se presenta, es la combinación de la plataforma Windows y cualquier otra, en lo que se refiere a la administración e interacción de los usuarios con los sistemas.
- Los manejadores de bases de datos relacionales están sustituyendo lo que es la utilización de archivos, debido a la facilidad que ofrecen para administrar información masiva, en este caso, no importa si éstos sistemas administradores son de código abierto (*opensource*) o si se adquieren a

través del pago de licencias, de igual manera son utilizados y muchas de las empresas combinan estos dos tipos de manejadores de bases de datos.

- La forma en la que se están desarrollando los sistemas de información en las organizaciones en Guatemala, indica que la mayoría se está inclinando por el desarrollo interno, en el caso de las empresas grandes y medianas; esto es, porque a largo plazo, es más rentable tener personal que esté optimizando y mejorando los sistemas ya desarrollados y no invertir cada vez en un sistema nuevo o en renovaciones más costosas al valor inicial de un sistema. En el caso de las empresas pequeñas, el *outsourcing* les trae mayor beneficio ya que la cantidad de procesos que automatizan es menor y en algunas ocasiones, su complejidad también, comparados con los procesos en las organizaciones grandes y medianas.
- En lo que se refiere a las metodologías de desarrollo de software, la tendencia es que cada vez más empresas adaptan una o varias metodologías de desarrollo (dependiendo de los requerimientos del sistema que se desarrolle, la cantidad de personas y el tiempo), lo que es un punto importante para poder determinar la calidad del software que se desarrolla en cada organización y por consiguiente, en nuestro país.
- La arquitectura sobre la que funcionan los sistemas integrados, en su mayoría utilizan la Arquitectura Orientada a Servicios (SOA), lo que implica la utilización y desarrollo de servicios Web y por consiguiente, un lenguaje estándar de comunicación, XML.

## 6.2 Expectativas

- De acuerdo a lo que se ha podido observar a través del estudio realizado, una de las expectativas es que las organizaciones puedan alcanzar un nivel de integración óptimo en sus sistemas en un período de tiempo relativamente corto (se podrían considerar cinco años o menos), esto, porque las herramientas que salen al mercado son cada vez más fáciles de administrar y son adaptables a las necesidades de cada empresa.
- Elevar el nivel de calidad de los sistemas que se desarrollan, esto se debe a que se están adoptando estándares para el desarrollo de software, además de la adquisición de herramientas que facilitan cada vez más el trabajo y reducen la cantidad de errores que puedan cometerse durante el desarrollo.



## CONCLUSIONES

1. Cuando se habla de integración de sistemas, se refiere a la conexión y mutua colaboración entre actividades o procesos, de manera que puedan unirse en forma coherente y poder funcionar como un todo, lo cual se exige para que pueda lograrse también, la integración de negocios.
2. Para poder seleccionar una metodología de desarrollo, es necesario tomar en cuenta aspectos como el contexto del proyecto, los recursos tanto técnicos como humanos, el tiempo del que se dispone para el desarrollo y el tamaño del proyecto, porque no existe una metodología universal sino solamente una variedad de metodologías que refuerzan distintos aspectos.
3. En Guatemala, la calidad de los sistemas que las empresas adquieren y desarrollan dentro del país, aumenta cada día, debido a que éste desarrollo se realiza bajo estándares reconocidos a nivel mundial y las empresas utilizan herramientas que reducen el tiempo y cantidad de errores que se le presenten a los desarrolladores.
4. La tendencia que se observa en la integración de sistemas en Guatemala, es que, todas las empresas, sin importar la actividad que realicen y sin importar tampoco, si pertenecen al sector público o privado, tienen una visión a corto o mediano plazo de realizar integración entre sus sistemas de información, lo que nos llevaría a obtener sistemas informáticos organizacionales más robustos y con ello, lograr un crecimiento más acelerado de las empresas y a la vez ofrecer servicios de mayor calidad.

5. El desarrollo de sistemas de información por parte de terceros, es una modalidad que se ha ido adoptando en su mayoría, por las empresas pequeñas, siendo el desarrollo interno la modalidad que están adoptando las grandes y medianas empresas, esto debido al balance que existe entre el costo y el beneficio que se obtiene al escoger una u otra opción y en algunos casos, mezclar ambas opciones. Por lo anterior, se podrá observar el aumento en la cantidad de empresas que incluyen entre sus actividades el desarrollo de software, aunque no sea su actividad económica principal.

## RECOMENDACIONES

1. Cuando se selecciona una tecnología para la integración de sistemas, se debe considerar la capacidad que ofrece y las necesidades de integración que tiene una organización ya que de esta forma, se podrán utilizar todas las facilidades que provee además de cumplir con los requerimientos establecidos por el nivel de integración que se desee alcanzar.
2. Al integrar sistemas sobre una arquitectura orientada a servicios, deben establecerse paquetes para los servicios de manera que puedan manejar la información que se intercambia entre los diferentes sistemas, estableciendo el tamaño más adecuado para que no exista una sobrecarga en el envío de paquetes y también para que no sean tan grandes que haya desperdicio de recursos.
3. Al momento de utilizar cualquier metodología de desarrollo y establecer una planificación para la realización de un proyecto, siempre deben tomarse en cuenta aspectos como la experiencia y la capacidad de los integrantes del equipo de desarrollo, a través métricas de proyectos desarrollados previamente; además, cuando existe un desequilibrio entre la cantidad de trabajo y las personas, debe hacerse una relación entre el tiempo que tomaría el desarrollo con las personas con las que se cuenta actualmente y el tiempo y costo que toma reclutar e integrar nuevo personal



## BIBLIOGRAFÍAS

### **1. Arquitectura Orientada a Servicios.**

[www.luismor.com/UtilizaciondePatronesParaLaConstrucciondeArquitecturasOrientadasAServicios.pdf](http://www.luismor.com/UtilizaciondePatronesParaLaConstrucciondeArquitecturasOrientadasAServicios.pdf) (10/10/2007)

[http://www.epidataconsulting.com/tikiwiki/tiki-read\\_article.php?articleId=44](http://www.epidataconsulting.com/tikiwiki/tiki-read_article.php?articleId=44)  
(07/12/2007)

[http://www.huibert-aalbers.com/IT\\_Insight/Spanish/PDF/ITI005Sp-SOA\(II\).pdf](http://www.huibert-aalbers.com/IT_Insight/Spanish/PDF/ITI005Sp-SOA(II).pdf) (02/02/2008)

### **2. Ciclos de vida de ingeniería de software.**

<http://www.naturastock.com/rsdotnet/iic3140/materia/cascadaymodelov.pdf>  
(13/10/2007)

### **3. Ciclo de vida del software.**

<http://alarcos.inf-cr.uclm.es/doc/ISOFTWAREI/Tema03.pdf> (05/11/2007)

### **4. Definición y características del software.**

[http://www.eici.ucm.cl/Academicos/R\\_Villarroel/descargas/ing\\_sw\\_1/ModeloSProcesoSoftware.pdf](http://www.eici.ucm.cl/Academicos/R_Villarroel/descargas/ing_sw_1/ModeloSProcesoSoftware.pdf) (03/02/2008)

**5. Desarrollo en espiral.**

[http://es.wikipedia.org/wiki/Desarrollo\\_en\\_espiral](http://es.wikipedia.org/wiki/Desarrollo_en_espiral) (17/01/2008)

**6. Desarrollo iterativo y creciente.**

[http://es.wikipedia.org/wiki/Desarrollo\\_iterativo\\_y\\_creciente](http://es.wikipedia.org/wiki/Desarrollo_iterativo_y_creciente) (03/03/2008)

**7. Gestión de procesos e integración de sistemas.**

<http://www.afi.es/infoanalistas/comun/mostrarFichero.asp?idContenido=725191&idSeccion=156467&usuario=TIFBrewery&password=TIFBrewery>  
(30/11/2007)

**8. Ingeniería de software.**

[http://es.wikipedia.org/wiki/Ingenier%C3%ADa\\_de\\_software](http://es.wikipedia.org/wiki/Ingenier%C3%ADa_de_software) (26/10/2007)

<http://delta.cs.cinvestav.mx/~pmejia/softeng/Cap1.ppt> (04/04/2008)

**9. Integración de aplicaciones.**

<http://www.edicomgroup.com/es/documentosEDI/EBIntegracionDeAplicacionesEd2.pdf> (25/11/2007)

**10. Integración de datos en la organización: Necesidades y soluciones**

[www.acis.org.co/fileadmin/Base\\_de\\_Conocimiento/XXV\\_Salon\\_de\\_Informatica/IntegracionDatos-JoseAbasolo.ppt](http://www.acis.org.co/fileadmin/Base_de_Conocimiento/XXV_Salon_de_Informatica/IntegracionDatos-JoseAbasolo.ppt) (15/12/2007)

### **11. Integración de sistemas.**

[http://prof.usb.ve/lmendoza/Documentos/PS-6117%20\(Teor%EDa\)/Teor%EDa%20PS6117%20Integraci%F3n%20de%20Sistemas.pdf](http://prof.usb.ve/lmendoza/Documentos/PS-6117%20(Teor%EDa)/Teor%EDa%20PS6117%20Integraci%F3n%20de%20Sistemas.pdf) (20/01/2008)

### **12. Introducción a la ingeniería de software.**

<http://kybele.escet.urjc.es/documentos/IS/%5BIS-2006-07%5DT1-INTRODUCCION.pdf> (07/05/2008)

[http://kybele.escet.urjc.es/documentos/ISG/%5BISG-2006-07%5DIntroduccion\\_ISG.pdf](http://kybele.escet.urjc.es/documentos/ISG/%5BISG-2006-07%5DIntroduccion_ISG.pdf) (23/04/2008)

### **13. Metodologías de desarrollo de software**

[http://is.umb.edu.co/aulaumb/file.php/33/METODOLOGIAS\\_DE\\_DESARROLLO\\_DE\\_SOFTWARE\\_V01.ppt](http://is.umb.edu.co/aulaumb/file.php/33/METODOLOGIAS_DE_DESARROLLO_DE_SOFTWARE_V01.ppt) (17/12/2007)

### **14. Modelo de prototipos.**

[http://es.wikipedia.org/wiki/Modelo\\_de\\_prototipos](http://es.wikipedia.org/wiki/Modelo_de_prototipos) (20/03/2008)

<http://www.sidar.org/recur/desdi/traduc/es/visitable/maner/Prototipado.htm> (09/02/2008)

**15. Orientaciones para la selección de tecnologías de integración de sistemas de software.**

<http://www.actea.es/C19/Tecnologia/Document%20Library/Integracion%20Sistemas%20Software.pdf> (03/05/2008)

**16. Proceso unificado de Rational (RUP)**

[http://es.wikipedia.org/wiki/Proceso\\_Unificado\\_de\\_Rational](http://es.wikipedia.org/wiki/Proceso_Unificado_de_Rational) (06/07/2008)

[http://www.informatizate.net/articulos/metodologias\\_de\\_desarrollo\\_de\\_softw\\_are\\_07062004.html](http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_softw_are_07062004.html) (03/02/2009)

<http://ivanex.wikidot.com/metodologia> (18/09/2008)

**17. Sistemas de información y los negocios**

[www.econ.unicen.edu.ar/academicas/catedras/sistemasdeinformacion/material/Tipificacion%20de%20los%20SI.ppt](http://www.econ.unicen.edu.ar/academicas/catedras/sistemasdeinformacion/material/Tipificacion%20de%20los%20SI.ppt) (13/10/2008)

**18. SOA como evolución natural de las soluciones de integración**

<http://desarrollo.datco.cl/MaterialPublico/EvolucionSOA.ppt> (04/02/2008)

**19. Tecnologías para el acceso a base de datos**

[http://es.wikipedia.org/wiki/Open\\_Database\\_Connectivity](http://es.wikipedia.org/wiki/Open_Database_Connectivity) (04/03/2009)

**20. The Software Experts / Quality Software Services.**

[http://www.the-software-experts.de/e\\_dta-sw-process.htm](http://www.the-software-experts.de/e_dta-sw-process.htm) (31/10/2008)

**21. Virtual Storage Access Method**

[http://es.wikipedia.org/wiki/Virtual\\_Storage\\_Access\\_Method](http://es.wikipedia.org/wiki/Virtual_Storage_Access_Method) (04/06/2008)

