



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

**INTEGRACIÓN DE NEGOCIOS,  
COMO INTEGRAR LOS SISTEMAS DE INFORMACIÓN**

**Yonny Rivahí Serrano Moya**  
**Asesorado por Ing. Roberto Marroquín**

**Guatemala, octubre de 2003**

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**INTEGRACIÓN DE NEGOCIOS,  
COMO INTEGRAR LOS SISTEMAS DE INFORMACIÓN**

TRABAJO DE GRADUACIÓN

PRESENTADO A JUNTA DIRECTIVA DE LA

FACULTAD DE INGENIERÍA

POR

**YONNY RIVAHÍ SERRANO MOYA**

ASESORADO POR ING. ROBERTO MARROQUÍN

AL CONFERÍRSELE EL TÍTULO DE

**INGENIERO EN CIENCIAS Y SISTEMAS**

GUATEMALA, OCTUBRE DE 2003

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA



### **NÓMINA DE JUNTA DIRECTIVA**

DECANO	Ing. Sydney Alexander Samuels Milson
VOCAL I	Ing. Murphy Olympo Paiz Recinos
VOCAL II	Lic. Amahán Sánchez Álvarez
VOCAL III	Ing. Julio David Galicia Celada
VOCAL IV	Br. Kenneth Issur Estrada Ruiz
VOCAL V	Br. Elisa Yazminda Vides Leiva
SECRETARIO	Ing. Pedro Antonio Aguilar Polanco

### **TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO**

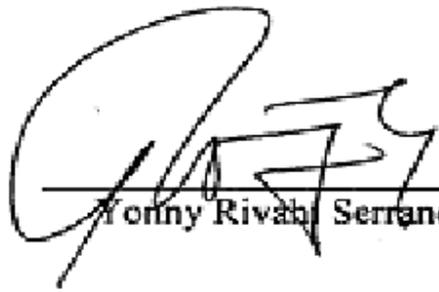
DECANO	Ing. Sydney Alexander Samuels Milson
EXAMINADORA	Inga. Virginia Victoria Tala Ayerdi
EXAMINADOR	Ing. Edgar Estuardo Santos Sutuj
EXAMINADOR	Ing. Alvaro Miguel Medrano Molina
SECRETARIO	Ing. Pedro Antonio Aguilar Polanco

**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:

**INTEGRACIÓN DE NEGOCIOS,  
COMO INTEGRAR LOS SISTEMAS DE INFORMACIÓN**

Tema que me fuera asignado por la Dirección de la Escuela de Ciencias y Sistemas con fecha de enero de 2001.



\_\_\_\_\_ Tony Riván Serrano Moya

## ÍNDICE GENERAL

<b>ÍNDICE DE ILUSTRACIONES</b> .....	V
<b>GLOSARIO</b> .....	VII
<b>RESUMEN</b> .....	XI
<b>OBJETIVOS</b> .....	XIII
<b>INTRODUCCIÓN</b> .....	XIV
<b>1. ¿QUÉ ES INTEGRACIÓN DE NEGOCIOS (<i>BUSINESS INTEGRATION</i>)?</b>	
1.1. Factores que influyen sobre la competitividad de una empresa . . . . .	1
1.2. Tecnología de la Información ( <i>Information Technology</i> ). . . . .	3
1.3. Integración de Negocios ( <i>Business Integration</i> ). . . . .	5
<b>2. ÁREAS A LAS QUE SE ORIENTA LA INTEGRACIÓN DE NEGOCIOS</b>	
2.1. Fusiones y adquisiciones . . . . .	7
2.2. Integración a la <i>Web</i> . . . . .	11
2.3. Manejo de la cadena de distribución . . . . .	12
2.4. Manejo de relaciones con clientes . . . . .	13
2.5. Planeación de recursos empresariales . . . . .	16
2.6. Procesamiento directo (STP) . . . . .	16
2.7. Puntos más importantes que se encuentran durante una integración .	17
<b>3. FORMAS DE IMPLEMENTAR UNA INTEGRACIÓN DE NEGOCIOS</b>	
3.1. La necesidad de una integración. . . . .	19
3.2. Técnicas para implementar la integración de negocios. . . . .	21
3.2.1. Soluciones rudimentarias . . . . .	21
3.2.2. Programación de comunicaciones. . . . .	23

3.2.3. Herramientas Especializadas .....	26
<b>4. HERRAMIENTAS ORIENTADAS A LA IMPLEMENTACIÓN DE LA INTEGRACIÓN DE NEGOCIOS</b>	
4.1. IBM <i>MQSeries</i> .....	33
4.1.1. Mensajes y colas .....	36
4.1.2. Gestor de colas ( <i>Queue Manager</i> ).....	41
4.1.3. Seguridad .....	44
4.1.4. Conexión de aplicaciones .....	46
4.1.5. Recuperación .....	49
4.1.6. Administración.....	52
4.1.7. Ambiente cliente servidor.....	55
4.2. <i>MQSeries Integrator</i> .....	58
4.3. IBM <i>MQSeries Workflow</i> .....	63
4.4. MSMQ (Microsoft <i>Message Queue Server</i> ).....	69
4.5. Consideraciones de Planeación.....	75
<b>5. ARQUITECTURA EN UN CASO REAL Y OTROS EJEMPLOS</b>	
5.1. Descripción del caso.....	79
5.2. Sistemas involucrados y relaciones entre aplicaciones.....	80
5.3. Arquitectura de comunicación .....	83
5.4. Configuración de <i>MQSeries Integrator</i> .....	84
5.5. Otros ejemplos reales .....	86
5.5.1. Una compañía distribuidora de artículos para el hogar. . . .	87
5.5.2. Un banco y una institución estatal.....	88
5.5.3. Tres sistemas internos de un banco.....	89
<b>CONCLUSIONES.....</b>	<b>91</b>

<b>RECOMENDACIONES.....</b>	<b>92</b>
<b>BIBLIOGRAFÍA.....</b>	<b>94</b>
<b>ANEXOS .....</b>	<b>97</b>

## ÍNDICE DE ILUSTRACIONES

### FIGURAS

1	Estructura y clasificación del <i>Middleware</i>	28
2	Dos aplicaciones intercambiando información	34
3	Representación de un mensaje	37
4	Esquema simple de comunicación entre dos gestores de colas	44
5	Componentes de un sistema	45
6	Cinco plataformas distintas	59
7	Cinco gestores de colas configurados utilizando un agente de mensajes	60
8	Módulos de comunicación entre un banco y una compañía de teléfonos	81
9	Esquema de comunicación utilizando <i>MQSeries</i>	83
10	Tres sistemas internos de un banco interconectados	90

## GLOSARIO

### *ActiveX*

Es el nombre que Microsoft le ha dado a un conjunto estratégico de herramientas y tecnologías de programación orientada a objetos. La tecnología principal es el COM (*Component Object Model*). El objeto principal que se crea cuando se está realizando un programa que corra en un ambiente *ActiveX*, es un programa auto-suficiente que puede correr en cualquier punto de la red *ActiveX* (compuesta por sistemas *Windows* y *Macintosh*). Este objeto principal es conocido como control *ActiveX*.

### **API (*Application Program Interface*)**

Es el método específico establecido por un sistema operativo o por una aplicación por medio del cual un desarrollador puede escribir un programa que pueda hacer requerimientos a ese mismo sistema operativo o aplicación.

### AS/400

Conocido ahora como *eServer iSeries/400*, es un servidor de tamaño medio diseñado por IBM para pequeñas empresas y para departamentos dentro de grandes empresas. Utiliza un microprocesador *powerPC* con su propio y reducido conjunto de instrucciones y su propio sistema operativo conocido como OS/400.

### **Call center (CRM)**

Un *call center* o CRM es un sitio central en donde una organización recibe y maneja las llamadas provenientes de

sus clientes, esta tarea es realizada generalmente con un sistema de información. El *call center* tiene la habilidad de manejar un volumen considerable de llamadas, al mismo tiempo resuelve las llamadas o las traslada a alguien calificado para hacerlo. Las compañías lo utilizan para proporcionar información y ayuda a sus clientes, para enviar catálogos, para realizar tele-mercadeo, etc.

### **Extranet**

Una extranet es una red privada que utiliza el protocolo de Internet y el sistema público de telecomunicaciones para, de una forma segura, compartir la información de negocios con proveedores, vendedores, asociados, clientes, etc. Puede ser vista como una parte de la intranet que se utiliza para enlazar a usuarios afuera de la empresa.

### **Formato de mensaje**

Es la estructura que tiene un mensaje y que permite representar al mensaje como un registro compuesto por uno o más campos de datos.

### **Intranet**

Una intranet es una red privada que existe dentro de una empresa. Puede estar formada por varias redes de área local enlazadas. Típicamente, una intranet incluye conexiones a través de una o más puertas de enlace (*gateways*). La meta principal de una intranet es compartir la información y los recursos de la empresa entre los empleados. También se utiliza para facilitar el trabajo en grupo y para realizar tele conferencias.

**IT (*Information Technology*) o Tecnología de la información**

Es un término que reúne todas las formas de tecnología utilizada para crear, almacenar, intercambiar y usar la información en cualquiera de sus formas (datos de negocio, archivos de voz, imágenes, documentos, etc.) Es la tecnología que hace posible el manejo de lo que se conoce como revolución de la información.

***Mainframe***

Es un término utilizado para referirse a una computadora de grandes dimensiones y capacidad, históricamente se asocia al *mainframe* con procesamiento centralizado más que con procesamiento distribuido. Esto significa que alrededor del *mainframe* usualmente se encontraba una red de terminales tontas.

**Manejo de la cadena de distribución**

Se refiere a la coordinación e integración de los flujos de materiales e información que existen de proveedores a fabricantes a vendedores a distribuidores y de ahí a los consumidores. Se dice que el objetivo principal de cualquier cadena de distribución efectiva es reducir el inventario.

**MAPI (*Messaging Application Program Interface*)**

Es una interface de Microsoft que permite enviar un correo electrónico desde una aplicación de *Windows* y adjuntarle el documento en el que se está trabajando a la nota electrónica.

**OS/2**

Es un sistema operativo de IBM para computadoras personales que inicialmente fue creado para proveer una alternativa al *Windows* de Microsoft.

<b>Plataforma</b>	Una plataforma es un sistema computacional en donde las aplicaciones pueden ejecutarse. <i>Windows 2000</i> y <i>Macintosh</i> son ejemplos de dos plataformas distintas para computadoras personales. La plataforma consiste básicamente en un sistema operativo que se encarga de coordinar los recursos de la computadora.
<b>Procesamiento <i>batch</i></b>	Un proceso <i>batch</i> es un programa que debe correr dentro de la computadora sin interacción inmediata del usuario. Como ejemplo de procesos <i>batch</i> se puede mencionar un trabajo de impresión. Algunos procesos <i>batch</i> son iniciados por un usuario del sistema o administrador y otros están programados para correr a cierta hora. Generalmente corren en <i>background</i> , es decir, sin mostrar en pantalla el estado del proceso.
<b>Reingeniería</b>	Es una modalidad (nueva) de la administración fundamentada en la premisa de que no son los productos, sino los procesos que los crean los que llevan a las empresas al éxito. Es la revisión fundamental y el rediseño radical de procesos para alcanzar mejoras espectaculares en medidas críticas y actuales de rendimiento, tales como costos, calidad, servicio y rapidez.
<b><i>Remote Procedure Calls</i> (RCP)</b>	Es el tipo de <i>middleware</i> más viejo y es también el más fácil de usar y entender. Proporcionan a los desarrolladores, la posibilidad de invocar a una función dentro de un programa y

obtener la ejecución de dicha función en una máquina remota.

***Socket***

Son puntos terminales a los que se conectan tanto el transmisor como el receptor para establecer un servicio de transporte de paquetes.

**TCP/IP (*Transmission Control Protocol/Internet Protocol*)**

Es el lenguaje de comunicación básico de Internet. Se utiliza también como un protocolo de comunicación en redes privadas.

**UNIX**

Es un sistema operativo que se originó en los Laboratorios Bell en 1969 como un sistema interactivo de multiprocesamiento. En 1974, UNIX se convirtió en el primer sistema operativo escrito en el lenguaje de programación C.

***WebSite***

Es una colección de archivos web (páginas) relacionados con un asunto o tema en particular. Dentro de estos archivos o páginas existe uno que constituye el inicio o *home page* del *website*. Esta página de inicio es la que aparece cuando se ingresa a la dirección de Internet de la compañía o individuo, a partir de ahí, de la página de inicio se tiene acceso a todas las otras páginas del sitio.

**XML (*Extensible Markup Language*)**

Es una forma flexible de crear formatos de información comunes y compartir la información y el formato dentro de la *world wide web*, en intranets, etc.



## RESUMEN

Para que una empresa pueda sobrevivir en el nuevo y competitivo mundo de los negocios, sin importar a que se dedique, es necesario que esté comunicada con sus proveedores, con sus canales de distribución, y con sus clientes. Esto le ayudará a responder de una manera rápida y efectiva a las necesidades del mercado. Sin embargo, lograr esto no es fácil, ya que si la interconexión entre diferentes sistemas operativos es algo muy complejo, la integración de negocios lo es aún mas y más cuando se sabe que hoy en día en el medio existen miles de plataformas de ejecución de *software* de diferente tipo, con sistemas operativos y de aplicación totalmente diferentes.

La integración de negocios es la colaboración entre diferentes sistemas de una o más empresas, lo que significa adaptar y conectar los sistemas existentes para que puedan intercambiar información en beneficio de los objetivos de la empresa. La integración debe permitir una comunicación rápida y flexible entre los sistemas.

La siguiente es una lista de situaciones (o necesidades) en las que una empresa puede encontrarse, haciendo necesario que sus sistemas de información se integren entre ellos mismos o con los sistemas de otras empresas.

- Fusiones y adquisiciones
- Integración a la web
- Agilizar el manejo de la cadena de distribución
- Mejorar el manejo de relaciones con clientes
- Planeación de recursos empresariales
- Mejorar el procesamiento directo

Existen varias formas para implementar una integración de negocios, entre ellas las más importantes son: soluciones rudimentarias, programación de comunicaciones y herramientas especializadas.

Entre las herramientas especializadas, la de mayor uso para implementar los proyectos de integración es la familia de productos *MQSeries* de IBM. Dentro de la familia se cuenta con la herramienta para el manejo de la comunicación por medio de mensajes, también existe un agente de mensajes o *message broker* que es capaz de resolver el reformato de mensajes y el enrutamiento inteligente. Además, uno de los productos de esta familia maneja flujos de procesos.

Las integraciones pueden efectuarse con los sistemas internos de una empresa o entre sistemas de dos o más empresas. Por ejemplo, dentro de una empresa pueden integrarse sus sistemas de inventario, de distribución y atención al cliente para que con el intercambio fluido de información se puedan optimizar los procesos. Algunas empresas que constantemente realizan proyectos de integración son los bancos, las empresas que prestan algún servicio (teléfonos, energía eléctrica, etc.), empresas transnacionales, instituciones estatales, etc.

## OBJETIVOS

- General

Dar a conocer y explicar qué es la integración de negocios, cuándo se debe implementar, qué herramientas existen para hacerlo y cómo se puede implementar satisfactoriamente.

- Específicos

1. Explicar detalladamente cada uno de los conceptos y/o fundamentos teóricos en los que se basa la integración de negocios.
2. Mostrar algunas de las herramientas desarrolladas para la puesta en marcha de la integración de negocios y explicar su funcionamiento.
3. Describir cuales son las areas y los puntos involucrados que hacen de la integración de negocios la solución para poder competir en el mercado actual.
4. Explicar cómo se puede aplicar el concepto de integración de negocios a un ejemplo real, haciendo el análisis y construyendo la arquitectura de colas de mensajes necesaria para el caso en particular.
5. Dar a conocer en detalle las características de una de las herramientas especializadas más importantes e ilustrar el funcionamiento para resolver la integración de negocios: *MQSeries*.

## INTRODUCCIÓN

Fuerzas como la competencia por la globalización, las alianzas de empresas, el comercio electrónico, el aumento de sofisticación de los consumidores y las presiones por reducir o mantener costos, son un desafío para los actuales modelos de negocio. Además, las nuevas oportunidades de negocio y las nuevas estrategias hacen de la integración de las aplicaciones de negocios un factor crítico para empresas que desean ser competitivas, y el desafío está en alcanzar la integración en el diverso entorno informático en el que ellas se desenvuelven.

Actualmente, el máximo desafío de la mayoría de estrategias de negocios como fusiones corporativas, procesos de reingeniería, optimización de la cadena de producción, manejo de la relación con el cliente y el comercio electrónico, depende en su totalidad de la manera en que la tecnología de la información los haga funcionar, especialmente cuando éstos están esparcidos alrededor del mundo. También, los encargados del desarrollo de aplicaciones, necesitan una manera más eficiente de extenderlas, ya sea que hayan sido o no, escritas con lenguajes de programación tradicional, o con los nuevos lenguajes de programación; de igual manera, corporaciones que jamás se imaginaron estar juntas, ahora necesitan estar unidas a través de sus redes de cómputo, y necesitan hacerlo de una manera fácil, que no solo permita conectar a los empleados, sino también a los clientes y proveedores de mismas.

En el presente trabajo se explica con detalle el significado de integración de negocios. Se muestra el por qué de la integración, es decir, qué fuerzas globales hacen que la integración sea una necesidad para las empresas. También se presentan los casos cuando la integración debe realizarse para mejorar la calidad de servicio, para hacer más eficiente la comunicación entre sistemas y para hacer frente con mayor fuerza a la competencia.

Luego de presentar el por qué y el cuándo de la integración de negocios, mostramos el cómo: se muestran varias formas de resolver la integración como programación de comunicaciones y herramientas especializadas, y se presentan las ventajas y desventajas de utilizarlas. En una sección completa se listan los pasos básicos a seguir cuando se está planificando una integración de negocios. Luego se describe en detalle una de las herramientas especializadas, una familia de productos que resuelve la integración, mostrando sus componentes y sus características más importantes.

Finalmente se presenta un ejemplo basado en un caso real, en donde se hace el análisis completo de la solución y se construye la arquitectura de aplicaciones y procesos involucrados en la integración. También se describen brevemente otros ejemplos reales para mostrar la diversidad de empresas que pueden verse empujadas hacia una integración con asociados de negocios.

## 1. ¿QUÉ ES INTEGRACIÓN DE NEGOCIOS (*BUSINESS INTEGRATION*)?

Gracias a la velocidad a la que cambia la tecnología, los mercados hoy en día cambian también a una gran velocidad. Por esta razón todas las empresas deben de responder de forma rápida y eficiente a esos cambios para poder sobrevivir y ser competitivas. Antes de entrar de lleno a lo que es integración de negocios, se debe de considerar cuales son los factores que una empresa debe de tomar en cuenta si quiere considerarse como una empresa competitiva.

### 1.1. Factores que influyen sobre la competitividad de una empresa

Existen algunos factores que son claves y que se deben de tomar en cuenta para poder sobrevivir en un entorno de negocios tan cambiante como el de hoy, uno de esos factores es la globalización. ¿Qué se entiende por globalización? La globalización es un fenómeno desarrollado en principio por las grandes empresas transnacionales, que tienen su casa matriz en algún país líder, económica y políticamente hablando, y un número  $x$  de sucursales en diferentes países alrededor del mundo.

Otras formas de globalización son también los bloques económicos, que pueden abarcar desde regiones hasta continentes enteros, como es el caso de la unión europea. También los tratados bilaterales de cooperación económica son un ejemplo de globalización, así pues, el fenómeno de la globalización ya sea del tipo económico o no, no es mas que tratar de formar, a través de la agrupación de ciertos elementos con las mismas características y/o necesidades, una sola entidad que pueda competir y desarrollarse en cualquier renglón de la vida humana, beneficiando por supuesto a cada uno de los elementos de ese todo.

Las fusiones y alianzas de empresas es otra tendencia hacia donde se está dirigiendo la economía mundial; aquí en Guatemala se puede observar que se han producido fusiones entre empresas bancarias. En Estados Unidos las alianzas de negocios entre IBM y *Cisco Systems*, la fusión entre *Compaq* y *Hewlett Packard*, la compra que hizo la empresa de telecomunicaciones MCI de su similar *Sprint*, formando así un nuevo monstruo de las telecomunicaciones llamado *WorldCom*, capaz de competir con AT&T, son ejemplos de dicha tendencia. Entonces, las fusiones y alianzas de negocios son mecanismos que usan las empresas para poder captar un mayor número de clientes, beneficios en la compra y distribución de productos y economías de escala, y por consiguiente un mayor número de utilidades, estando en posibilidad de competir con empresas o bloques de empresas más fuertes y así lograr sobrevivir en el mercado.

Pero se debe de tomar en cuenta que no sólo con estar atentas a la globalización, las alianzas y las fusiones, las empresas de hoy pueden considerar que son competitivas y que pueden sobrevivir, no se deben de dejar a un lado los factores que han sido preocupación de siempre, como lo son dar un buen servicio al cliente, que las cadenas de producción y distribución estén perfectamente controladas, para poder así mantener o reducir costos y aumentar utilidades. Porque no se deben descuidar estos factores, por ejemplo, en el caso de una fábrica de zapatos si descuida la eficiencia y los tiempos en su cadena de producción, lo más probable es que esto le repercuta en retrasos en la entrega del producto final, no obstante si su cadena de distribución es deficiente, provocando así cuantiosas pérdidas poniendo en riesgo su existencia.

Entonces, se necesita tener toda la información necesaria acerca de qué proveedores son más rápidos y la calidad de sus productos, y si en algún momento un proveedor no puede responder a alguna de las necesidades, que alternativas están disponibles. De igual manera sucede con las rutas y líneas de distribución.

La atención que se debe de prestar a los clientes es, hoy en día, un factor determinante, ya que gracias a la Internet y otros medios de comunicación, los clientes actuales y los potenciales saben cómo, cuándo, en dónde y a qué precios pueden comprar. Esto obliga a que se tenga una buena relación con los clientes actuales y se conozca sus preferencias y tendencias; y en el caso de los clientes potenciales se conozca la mejor manera de llegar a ellos y así poder cumplir de forma eficiente con los requerimientos del mercado que se quiere abarcar.

En todos los casos anteriores se ha mencionado la importancia de ciertos factores que influyen sobre la competitividad de las empresas, pero debe de observarse que existe un factor común implícito que es de suma importancia, la información, y específicamente a compartir la información. Cualquiera que sea la situación en que la empresa se encuentre, no importando si está fusionándose, haciendo una alianza, mejorando sus cadenas de producción y distribución, etc., debe de compartir información, y debe de estar preparada para poder compartirla y obtenerla de forma rápida y eficiente, para que cualquiera de las estrategias que emprenda tenga el éxito esperado.

## **1.2. Tecnología de la información (*Information Technology*)**

Desde que el hombre lleva registros de sucesos, recopila información sobre personas, accidentes geográficos, etc., ha utilizado diferentes técnicas y materiales para almacenar dicha información, pero el uso de la computadora ha hecho más eficiente y sofisticada la forma en que se almacena la información, no importando de que clase sea.

En general, la tecnología de la información se refiere a la creación, recopilación, procesamiento, almacenamiento y distribución de la información, y de los procesos y dispositivos que hacen todo eso posible. La idea que está detrás de la tecnología de la información es hacer la vida más eficiente.

La tecnología de la información permite procesar y convertir numerosos datos que por sí solos no significan nada, en información valiosa y con algún tipo de significado; permite reciclar información que ya ha sido procesada, y utilizarla en otro paso del proceso, permite empacar información colocándola en un formato fácil de entender, atractivo y de mayor utilidad.

Para el caso de la integración de negocios se tomará el punto de vista de los sistemas de información y se acordará que la tecnología de la información se refiere al desarrollo y utilización de aplicaciones y sistemas, que han sido creados para cubrir los negocios y/o necesidades completas de una empresa. También, cada vez que se utilice el término tecnología de la información o IT por sus siglas en inglés, *Information Technology*, se sabrá que dicho término tiene que ver con cosas como.

- Qué tipo de redes de computadoras se deberían de usar dentro de la empresa, o si se necesita alguno.
- Qué sistemas operativos se debieran de usar, para la administración de las bases de datos de clientes, datos financieros, modelos de negocio, etc., dentro de una empresa.
- Qué clase o tipo de información se debe de colocar dentro de los sistemas, y cómo debiera de ser compartida dentro de la compañía.
- De qué manera se deben de interconectar las diferentes áreas de la compañía para tener un eficiente intercambio de información.
- De qué forma se pueden usar las computadoras para implementar un buen programa de servicio al cliente, aumentar la velocidad de ejecución del negocio y aumentar la eficiencia del mismo.

### **1.3. Integración de negocios (*Business Integration*)**

Dos grandes compañías de ahorro y préstamo cancelan su fusión corporativa cuando descubren que sus sistemas de IT son incompatibles. Una fábrica de materiales eléctricos adquiere un sistema de administración de recursos de la empresa y se da cuenta que la integración de éste es más cara que su nueva adquisición y un fabricante de automóviles que invierte en un costoso *website*, y empieza a perder clientes, gracias a que su departamento de IT no integra apropiadamente su *website* con las aplicaciones que controlan sus pedidos y facturación, son un pequeño ejemplo de la cantidad de oportunidades de negocio que se pueden perder, gracias a las dificultades que implica la integración de los sistemas de IT.

Entonces, la resolución exitosa de cualquiera de las estrategias de negocio a las que se menciona anteriormente, y que se mencionan como factores de competitividad en la primera sección de este capítulo, depende de la colaboración entre los diferentes sistemas de la empresa, y esta colaboración significa, adaptar y conectar los sistemas existentes en beneficio de los nuevos alcances de la empresa, haciéndolo de una forma rápida y flexible. Por lo tanto, cuando todos los sistemas de un negocio pueden colaborar de esta manera, se puede decir que existe la integración de negocios (*Business Integration*).



## **2. ÁREAS A LAS QUE SE ORIENTA LA INTEGRACIÓN DE NEGOCIOS**

1. Fusiones y adquisiciones
2. Integración a la *web*
3. Manejo de la cadena de distribución
4. Manejo de relaciones con clientes
5. Planeación de recursos empresariales
6. Procesamiento directo

### **2.1. Fusiones y adquisiciones**

“Los estudios de mercado han demostrado que dos de cada tres fusiones no funcionan... en particular, la mayoría de las fusiones más recientes han fracasado debido a que no planificaron con anticipación la integración de sus sistemas de informática”

(The Economist, *enero 1, 1999*).

Se ha producido en las últimas décadas una gran cantidad de fusiones y adquisiciones como resultado de los cambios radicales que están ocurriendo en los modelos empresariales tradicionales. Las causas principales de estos cambios son la globalización, la competencia, clientes mejor informados, velocidad de transformaciones y el comercio electrónico. Las fusiones y adquisiciones se han convertido en la herramienta empresarial estándar para lograr un crecimiento más rápido, ampliar las capacidades, establecer una presencia global y aumentar la participación en el mercado. La consolidación, la convergencia y la globalización son los tres factores principales que impulsan el mercado de fusiones y adquisiciones.

- **Consolidación:** las fusiones y adquisiciones corporativas tienen como objetivo aumentar la participación de mercado y lograr acceso a bases de clientes específicas, habilidades especiales, clientes, tecnologías y productos.
- **Convergencia:** la fusión entre empresas de diferentes sectores se lleva a cabo para lograr mayores economías de escala, aumentar oportunidades de venta o abarcar un mercado más grande.
- **Globalización:** las adquisiciones corporativas se utilizan para establecer una presencia internacional, ingresar a nuevos mercados o cerrarle el paso a la competencia.

Entonces, cuando se produce una fusión corporativa, el objetivo principal de las empresas involucradas es aprovechar la fortaleza que representa la unión con las otras. La relación puede alcanzar un nivel muy profundo, como la integración o consolidación de los procesos, recursos e, incluso, las culturas empresariales. El reto principal, muchas veces, es integrar y alinear los sistemas de tecnología informática con la estrategia futura de negocio. Si la fusión o adquisición es un éxito, se generan oportunidades para comercializar una cartera de productos más amplia, ofrecer mejor servicio y reducir costos. La fusión puede fallar si las empresas participantes no logran integrar sus sistemas de informática y muy probablemente, perderían todos los beneficios de la fusión. Los efectos de una fusión fallida podrían ser: perder a ejecutivos claves, a clientes importantes, la confianza de los accionistas, etc.

La integración de los sistemas y redes de las empresas debe ser rápida y segura para que sea exitosa. Esta debe ser una de las consideraciones primordiales, incluso mucho antes de firmar cualquier acuerdo y se debe a la creciente dependencia que las empresas tienen en sus sistemas de informática.

La integración y alineación de dichos sistemas con los procesos y estrategias empresariales posteriores a la fusión, son los factores claves del éxito o del fracaso del proyecto.

Aspectos claves a tomar en cuenta con respecto al impacto que tiene la integración de la tecnología informática.

- **Tiempo y costo.** La integración requiere tiempo, el tiempo que no se tiene en la mayoría de proyectos de fusiones y adquisiciones. Las operaciones comerciales de rutina no pueden verse comprometidas por problemas con los sistemas de informática, particularmente aquéllos que soportan las actividades de atención al cliente.  
El costo constituye también un factor principal, debido a que uno de los objetivos más importantes de este tipo de proyectos es aumentar el valor de los accionistas. Por lo tanto, cualquier costo adicional asociado a problemas de integración debe ser eliminado.
- **Percepciones, internas y externas.** Se refiere aquí a la confianza de los accionistas, inversionistas, clientes, asociados de negocios, gerentes y personal. Cada fusión y adquisición debe ser estudiada cuidadosamente para asegurar su viabilidad y habilidad de satisfacer los objetivos de la misma.
- **El valor de las inversiones existentes en tecnología informática.** Es esencial preservar la inversión existente en tecnología informática, los activos involucrados en la tecnología informática pueden ser tangibles (por ejemplo, *software* y *hardware*) e intangibles (por ejemplo, bases de datos de clientes).

- **La integración de sistemas de informática diferentes.** La información es uno de los recursos más importantes de cualquier organización. Es esencial compartirla y utilizarla en todas las áreas de la empresa y, en una fusión o adquisición, con las áreas procedentes de las otras empresas involucradas.
- **Experiencia en tecnología informática.** Las empresas poseen una gran inversión oculta en tecnología: la experiencia y el conocimiento del personal que desarrolla, opera y utiliza los sistemas de informática. Cuando se lleva a cabo una fusión o adquisición, esta base de conocimiento intelectual se ve amenazada de dos maneras. En primer lugar, si los sistemas no pueden integrarse exitosamente, se perderá el valor del conocimiento asociado con su uso. En segundo lugar, si la integración lograda no ofrece el rendimiento requerido, los dos sistemas pueden caer en desuso como resultado de la insatisfacción de los usuarios. Si esto ocurre, se perderá la experiencia que forma parte sustancial de la calidad del servicio de la empresa.
- **Diferencias culturales y de procedimiento.** Hace pocos años, la mayoría de las empresas grandes utilizaban una tecnología de *mainframe*, la cual estaba basada en una cantidad pequeña de estándares aceptados. Hoy en día, debido al ritmo de cambios sin precedentes que atraviesa el ambiente de tecnología informática, el único estándar es poseer múltiples plataformas, sistemas operativos, tecnologías de comunicación y protocolos, que proporcionan soporte a una amplia variedad de ambientes de bases de datos, aplicaciones y computadoras. La integración de estos diversos ambientes en forma rápida y sin complicaciones durante las fusiones y adquisiciones, puede demorar mucho tiempo y resultar extremadamente costoso. Como si esto fuera poco, la integración debe ser flexible para adaptarse a los cambios futuros generados por el ambiente posterior a la fusión o adquisición.

## 2.2. Integración a la web

Se supone que una empresa quiere ofrecer un nuevo rango de servicios a través de Internet. Para esto deberá diseñar un sitio *web* que incluya, por ejemplo, un catálogo de productos, una herramienta de selección de productos, un sistema de órdenes en línea y un sistema de protección de los datos recibidos del cliente. Además, se debería integrar el sitio *web* a los sistemas de inventario, distribución, contabilidad y a todos los otros sistemas involucrados en una venta.

Diseñar un sitio que posea estas capacidades implica el desarrollo de nuevas aplicaciones, las interfaces para que éstas trabajen con las aplicaciones existentes, la integración de un amplio y diverso conjunto de aplicaciones al sitio *web*, así como la conexión entre el servidor *web* y los sistemas de computación de la empresa. También es importante tener en cuenta que las páginas del sitio probablemente estarán compuestas de varios archivos y que será necesario desarrollar un sistema que permita manejar el crecimiento del sitio.

Estos son los aspectos más importantes que la empresa ya posee y que pueden ser utilizados para adaptarse mejor al mundo del comercio electrónico.

- Experiencia y conocimiento del mercado
- Base de clientes
- Sistemas comprobados que se han ido perfeccionando y que podrían adaptarse a los sistemas de la *web*

Y, estos son los aspectos que deben considerarse para establecer una presencia en la *web* de alta calidad y con servicio completo.

- Información precisa y dinámica. La solución a este requisito es conectar los servidores de *web* con los sistemas internos que proporcionan información actualizada de los productos y servicios de la empresa.
- Transacciones electrónicas en tiempo real.
- Servicio personalizado, para esto es necesario tener la capacidad de retener, procesar, y ofrecer información clave durante las visitas de los clientes con conexiones seguras y rápidas a las bases de datos.
- Servicio con excelente tiempo de respuesta.
- Transacciones seguras.
- Mejoras continuas.

### **2.3. Manejo de la cadena de distribución**

Los productos llegan a las manos de los clientes gracias a la cadena de distribución de la empresa. Los beneficios de una cadena de distribución eficiente son, por ejemplo, las reducciones en el inventario, flujo de caja más eficientes, aumento en la satisfacción del cliente, mayor control de los procesos de inventario y fabricación. Para mejorar la cadena de distribución es necesario enlazar las plataformas o sistemas internos, mejorar las redes a nivel internacional y agilizar el acceso a los datos de los clientes. El rendimiento de la empresa mejora si se agilizan los trámites internos y se trabaja en conjunto con los asociados de negocio.

La cadena de distribución esta formada, básicamente, por los siguientes eslabones:

- Pronóstico de ventas
- Planificación de la producción
- Estrategias de abastecimiento a las sucursales y centros de distribución
- Planificación y optimización de la distribución (armado de pedidos y rutas óptimas)

Los sistemas involucrados en cada una de estas fases deben enlazarse para que intercambien información segura y rápidamente. Al lograrlo se estará construyendo una cadena de distribución más eficiente.

## **2.4. Manejo de relaciones con clientes**

Los clientes no siempre saben lo que quieren. Con un nivel de servicio superior lo que se busca es ayudar al cliente a examinar todas sus opciones y proporcionarle todos los recursos necesarios de la mejor forma posible. Para lograrlo, la empresa debe conocer los requerimientos de los clientes. Los representantes de servicio deben tener acceso inmediato en sus computadoras a las aplicaciones desarrolladas basándose en la experiencia recopilada del personal de ventas.

La información histórica que representa la experiencia del personal de ventas y servicios generalmente se encontrará dispersa en sistemas distintos. La información histórica que se menciona es, por ejemplo, perfiles de los clientes, productos complementarios, opciones que ofrece la empresa, etc. La integración de los sistemas de servicio al cliente proporciona la solución para las necesidades de recopilación, disponibilidad y acceso a la información histórica. Se debe establecer una infraestructura tecnológica que promueva la integración e interoperabilidad entre todos los sistemas de la empresa.

Las siguientes son algunas posibilidades que podrían utilizarse para mejorar el servicio al cliente:

- Utilizar diferentes medios para proporcionar servicio al cliente, como por ejemplo las unidades de respuesta por voz, centros de soporte, sucursales automatizadas, sitios web, quioscos y cajeros de autoservicio, así como sucursales y representantes de ventas. Se puede proporcionar el mismo nivel de servicio y respuesta, en forma consistente, a través de cada uno de estos vehículos.
- Comprender a fondo las necesidades de los clientes, basándose en un perfil completo de su relación actual y pasada con los mismos.
- Proporcionar el nivel más alto de servicio a los clientes más rentables. Asignar personal de acuerdo con el nivel de experiencia requerido para responder cada consulta, de forma que pueda utilizar a los representantes de servicio con mayor experiencia para atender a los clientes más valiosos.
- Aprovechar la experiencia de los mejores representantes de ventas de la empresa. Crear una base de conocimientos y experiencia y ponerla a disposición de todo el personal.

Los clientes perciben a la empresa como una única entidad, no como un grupo de funciones distintas. Pero, por lo general, las diferentes áreas de negocio están encerradas dentro de sus propias líneas de producto.

Para desarrollar una buena relación con los clientes, los sistemas centrales (los mismos que ejecutan los procesos de negocio fundamentales) necesitan interconectarse con los nuevos sistemas de atención al cliente, tales como las unidades de respuesta por voz, centro de soporte, sistemas automatizados, sitios web y quioscos de autoservicio. Cada nueva interfaz que se desarrolle para el cliente tendrá que integrarse a los sistemas centrales de la empresa.

Los elementos que se presentan a continuación forman parte de un esquema que permite integrar los sistemas de servicio al cliente.

- Toda la información del cliente, aunque esté almacenada en diferentes bases de datos y en múltiples sistemas, debe presentarse uniformemente, en una forma ordenada y completa.
- Es necesario un *software* robusto y confiable que permita interconectar las aplicaciones claves de negocio e intercambiar información entre ellos. También se necesita actualizar, reutilizar y desarrollar aplicaciones para nuevos canales, tales como la web.
- Las aplicaciones que se desarrollen deben correr en cualquier sistema.
- Automatización de los procesos de negocio. Los procesos automatizados funcionan con mayor rapidez y mantienen a los clientes satisfechos debido a que la información está a su alcance instantáneamente.
- También es necesaria la administración de todos los elementos desde un único punto.

## **2.5. Planeación de recursos empresariales**

Para conseguir una solución totalmente integrada es necesario que los sistemas internos estén integrados entre sí y con los sistemas de los proveedores de servicios y materia prima. La integración de negocios tiene como principal objetivo proporcionar una solución totalmente integrada, rápida y con riesgos mínimos. Para lograrlo es necesario reunir las operaciones fragmentadas, compartir información común entre las aplicaciones que la utilizan y enlazar los sistemas para facilitar el flujo de datos. Esto permite reducir el tiempo de respuesta a las demandas de los clientes, eliminar la duplicación de procesos, reducir el tiempo de espera y los errores en el traslado de información.

Con la planeación de los recursos empresariales se obtiene una visión de los procesos, prácticas y procedimientos de la empresa y, las soluciones para la integración de negocios permiten implementar los cambios necesarios de una forma rápida, eficiente y con riesgos mínimos.

## **2.6. Procesamiento directo (STP)**

El objetivo del procesamiento directo es reducir los costos que resultan de los retrasos en el procesamiento y los errores humanos durante las transacciones. Esto se lleva a cabo mediante la automatización del flujo de la información, desde el momento en que se reciben los datos hasta la confirmación de la transacción, a fin de que el tiempo de respuesta sea suficientemente rápido para ofrecer sustanciales beneficios operacionales. Dentro del intervalo que inicia cuando se reciben los datos y termina cuando se confirma el procesamiento de la orden, existe más de media docena de pasos necesarios para concretar la transacción.

Una empresa con cero latencia es aquella en la que el retraso entre cada paso es equivalente a cero. Sin embargo, los sistemas encargados de efectuar cada uno de los pasos son distintos y, muchas veces, son sistemas que pertenecen a empresas intermediarias (por ejemplo, en una institución financiera, las empresas intermediarias podrían ser corredores de bolsa, instituciones estatales reguladoras, etc.). Un esquema ideal para conseguir cero latencia requiere una integración estrecha entre los sistemas involucrados en cada paso de la transacción.

Al reducir el tiempo para procesar y concretar las transacciones se aumenta la capacidad de tráfico. Con la integración de los sistemas involucrados en las transacciones se reducen los riesgos financieros que representan las fallas operacionales, los retrasos de procesamiento y las equivocaciones del personal.

## **2.7. Puntos más importantes que se encuentran durante una integración**

- El desarrollo de un único sistema corporativo a partir de dos o más arquitecturas de sistemas principales, tales como la implantación de soluciones de planeación de recursos empresariales o de una cadena de distribución altamente integrada, sin afectar la operación del negocio.
- La combinación de los recursos de informática en forma precisa, segura y eficiente en costos para preservar el valor de la fusión o adquisición.
- La fusión de arquitecturas incompatibles sin duplicaciones ni redundancias costosas.

- Desplazarse rápidamente para integrar sistemas de tecnología informática, con el fin de prevenir que falte alguna capacidad tecnológica que pueda obstaculizar la fusión o adquisición y perjudicar el rendimiento de la empresa.
- La integración de los procesos empresariales de atención al cliente, para crear un solo ambiente operacional que permita preservar la calidad del servicio y proporcionar un panorama total del cliente en toda la empresa.
- La integración de diversos sistemas que sirven para un mismo propósito, con el fin de transformarlos en intranets corporativas de correo electrónico (*e-mail*) o extranets para clientes y socios.
- La incorporación de flexibilidad a las arquitecturas integradas, con el fin de implantar un sistema que se adapte a futuros desarrollos, así como a las necesidades actuales de la empresa.

### **3. FORMAS DE IMPLEMENTAR UNA INTEGRACIÓN DE NEGOCIOS**

Teniendo una idea clara sobre a qué se refiere la integración de negocios y las áreas hacia donde se puede enfocar, deben analizarse las diferentes formas de realizarla, en este capítulo se centrará la atención en algunas de las formas a las que se podría recurrir para la puesta en marcha de este concepto y las implicaciones y consecuencias que éstas conllevan. Antes de empezar a detallar cada una de las alternativas, es necesario plantear un ejemplo de la vida real en el que se podría presentar la necesidad de una integración.

#### **3.1. La necesidad de una integración**

En el medio se está manifestando un fenómeno que se puede tomar como un clásico ejemplo de necesidad de integración, éste, a simple vista, daría la impresión de ser un fenómeno comercial, económico, o como un efecto de la corriente conocida como globalización; esto es la fuerte competencia de la telefonía móvil celular que se ha producido en los últimos años en nuestro país.

Desde la venta de la empresa de telecomunicaciones del estado y la apertura para la entrada de empresas transnacionales, el mercado de la telefonía móvil a experimentado una dura guerra entre las empresas que compiten en el mismo, desde anuncios de televisión, de radio y escritos, hasta ofertas fabulosas de planes de pago con "x" minutos de aire, promociones, rebajas en los precios, etc. Estas empresas han implementado muchas estrategias para captar la mayor cantidad de clientes posible.

De esta dura y vertiginosa competencia, afortunadamente los beneficiados son los consumidores, teniendo éstos un sin número de opciones de donde escoger, pero, ¿Qué parte de este negocio de telefonía podría ser el punto clave para que los clientes decidieran estar con X o Y compañía? La respuesta a esta interrogante se podría obtener observando de forma detenida el sistema de cobros de la empresa, que es un tema que cuando está bien controlado pasa desapercibido, pero cuando no se tiene el control sobre él, puede ser el causante de que los clientes decidan abandonar una empresa.

Entonces, para una empresa de esta naturaleza su sistema de cobros puede representar una necesidad de integración de negocios. Desde el punto de vista de planeación estratégica y de negocios en sí, una empresa de telefonía móvil no puede centralizar su sistema de cobros, ya que en el lugar escogido para esto, se formarían enormes colas y el sistema se haría ineficiente, de tal forma que los clientes estarían insatisfechos con el servicio.

En contraparte, si la empresa en cuestión está compitiendo en un mercado tan exigente, no puede darse el lujo de abrir una gran cantidad de centros de cobro, ya que esto podría provocarle pérdidas desde el punto de vista que tendría que invertir en equipo y personal, en lugares que en un corto tiempo se tendrían que cerrar. Más bien, la solución a esta necesidad es aprovechar la infraestructura que ofrecen empresas como los bancos del sistema (1500 agencias), supermercados, gasolineras, etc., los cuales ya poseen sistemas especializados en cobros y captaciones de dinero.

De lo anterior sale la pregunta siguiente; ¿de qué forma una empresa de telefonía puede trabajar en conjunto con el sistema bancario nacional? Bueno, en los párrafos anteriores se mencionó que el sistema de cobros de una telefónica puede ser un punto clave para la captación y/o retención de su clientela, entonces, la telefónica podría evitarse los problemas de colas de un centro de cobros centralizado, y el problema de la inversión en la apertura masiva de centros de cobro, haciendo una alianza estratégica con alguno de los bancos del sistema.

Esta alianza no sólo beneficiaría a la compañía telefónica, sino que de cierta manera también beneficiaría al banco, ya que los usuarios por el deseo de centralizar sus operaciones bancarias podrían solicitar otra clase de servicios.

Desde el punto de vista de negocios está claro el beneficio, pero ahora la barrera tecnológica debe ser superada y no convertirse en la limitante. Entonces, ¿cuál sería el trabajo ahora?, el trabajo sería realizar la integración de negocios, haciendo que los sistemas involucrados, tanto de la telefónica como del banco trabajen en conjunto para producir los resultados esperados.

### **3.2. Técnicas para implementar la integración de negocios**

Existen varias formas de poder implementar una integración de negocios, pero aquí se ha tratado de encerrar las diferentes soluciones, en tres grupos.

- Las soluciones rudimentarias
- Programación de comunicaciones
- Herramientas especializadas

### 3.2.1. Soluciones rudimentarias

Al hacer uso de la palabra rudimentaria, no se hace con el fin de menospreciar la metodología de realizar un proceso, o la forma en que una empresa administra su tecnología de la información, si no que lo que se hace con el fin de clasificar algunas metodologías en particular que se utilizan en el caso de integración de negocios.

Para un mejor entendimiento de esta forma de implementar la integración, se refiere al ejemplo que se menciona en la sección anterior, la integración del sistema de cobro de una telefónica con el sistema de caja de una institución bancaria. En este caso, se sabe que la forma en que trabajarían en conjunto estos sistemas sería el siguiente: los clientes de la telefónica deberían de poder pagar sus cuotas mensuales en una agencia bancaria; este pago que se recibe a través de un cajero de agencia bancaria y se aplica en el sistema central del banco, debe de reflejarse también en el sistema de cómputo de la telefónica.

Entonces, ¿qué procedería después de efectuado el pago?. El siguiente paso debiera de ser que el banco le proporcione esa información a la telefónica, para que ésta pudiera aplicar los pagos y tener actualizada su información y prestar un buen servicio a sus clientes. Una solución a esto podría ser que la información a compartir se enviara a través de un fax, el cual contendría información con una copia de cada uno de los pagos que los cajeros del banco ha recibido durante el día, este fax sería recibido en la telefónica al final del día para poder redigitar la información.

Otra forma de resolver la integración podría ser un método de generación de archivos conocido como *trigger file*.

Este método sostiene que se puede generar un archivo plano, de texto o binario, para que se procese la información; esto quiere decir que la transacción de pago de teléfono hecha en una estación de caja del banco, pudiera generar registros o información a un archivo plano, para que al cabo de cierto tiempo o al final del día éste fuese enviado a la telefónica. Este envío puede realizarse por medio de un correo electrónico o por medio del envío de medios magnéticos (discos flexibles); el archivo sería recibido y procesado de una forma que se conoce como procesamiento en lotes o *batch*, para alimentar los sistemas de cómputo de la telefónica y mantener así actualizada la información.

Pero, ¿qué pasa con estas dos formas de integrar los sistemas involucrados? Las dos soluciones son fáciles y rápidas de implementar, pero ¿qué pasa con el tiempo de respuesta si el cliente quiere que se le active su teléfono en el momento en el que él ha realizado el pago?, ¿qué pasa con el manejo de errores y excepciones?, y si el cajero digitó mal el número del teléfono y se hizo el abono a la cuenta de otra persona, que pasa si el número de teléfono que se digitó no existe y no se le puede advertir de eso al cajero.

Entonces, estas formas de implementar o poner en marcha una integración de negocios son fáciles y prácticas pero ineficientes. A muy corto plazo la integración tendría que pagar un alto precio en el sentido que la solución es lenta e ineficiente y que en lugar de dar beneficios provocaría problemas en el manejo de los cobros a clientes, en el caso de la telefónica, y deterioraría la imagen del servicio que da la empresa, en el caso del banco. Es por estas razones que en este documento se llama a ese tipo de soluciones, las soluciones rudimentarias.

### **3.2.2. Programación de comunicaciones**

Cuando dos o más empresas deciden formar una alianza estratégica o fusionarse en busca de la obtención de beneficios mutuos, indiscutiblemente necesitarán compartir información; para lograrlo será necesario poner a trabajar en conjunto los sistemas de IT de cada uno de los miembros de la alianza, para alcanzar los resultados deseados. La integración de sistemas que se desea realizar puede ponerse en marcha utilizando los conocimientos de expertos en comunicaciones, aplicando sus conocimientos y experiencia en el desarrollo de una solución efectiva.

Entonces, cuando se hace referencia a programación de comunicaciones se está hablando de una técnica en la que se invierte bastante cantidad de tiempo en el análisis del problema, recolección de información y en el diseño y programación de las aplicaciones que van a realizar ciertas tareas sobre diferentes sistemas, sobre diferentes o similares protocolos de comunicación.

En el caso de la compañía telefónica y el sistema de caja de un banco, se sabe que la necesidad existente consiste en que los usuarios del servicio telefónico puedan hacer sus pagos a través de las agencias de un banco. Para poder llevar a cabo esta integración usando la programación de comunicaciones, primero se debe de determinar cuál será el proceso y cuál será el flujo de la información a procesar, por ejemplo, se podría llegar al acuerdo entre las partes que el proceso sería de la siguiente forma: primero un cliente de la telefónica llega a la agencia bancaria a realizar el pago de servicio telefónico, el cajero recibe el dinero y los datos, los procesa en su terminal dentro del sistema de caja; la información primeramente filtrada por el sistema local del cajero, viaja al sistema central del banco, el sistema central lo analiza y se da cuenta que es un pago de teléfono y lo envía a través de una línea de comunicaciones al sistema central de la telefónica.

En la empresa de telefonía se procesa la información y según su resultado se envía su aceptación al sistema del banco, o se envía la no-aceptación con el respectivo mensaje de error.

En el proceso anterior, la programación de comunicaciones entraría a funcionar en la parte del intercambio de información entre los sistemas centrales de cómputo del banco y la telefónica; y eso representa que se necesita saber qué protocolos de comunicación existen en los diferentes equipos de cómputo y con qué capacidades se cuenta, por ejemplo, si el sistema central del banco es un *mainframe* y el de la telefónica es un servidor Unix. Se necesita saber con qué protocolos de comunicación están trabajando, cuáles pueden soportar, se necesita decidir cuáles se van a usar y definir los mensajes de comunicación que se van a usar; lo anterior se refiere a:

- a. El caso más simple, si se sabe que el servidor Unix soporta el protocolo de comunicaciones TCP/IP, y el *mainframe* también lo soporta, la tarea en este caso consistiría en construir los aplicativos para los dos servidores, utilizando la programación de *sockets* que provee TCP/IP para comunicar dispositivos que trabajan sobre este modelo de comunicación.
- b. El segundo caso, y un poco más complicado que el anterior, sería que teniendo la capacidad de soportar TCP/IP del servidor Unix, el *mainframe* sólo pudiera soportar el esquema SNA de comunicaciones; en tal caso el objetivo de la programación de comunicaciones sería tratar de comunicar los dos sistemas usando un dispositivo intermedio conocido como puente *bridge*, que tomara la información de un lado, la transformara y la trasladara al otro. Esto quiere decir que para desarrollar los aplicativos en el *mainframe*, se podrían utilizar los servicios APPC que provee SNA, y para los aplicativos del servidor Unix se utilizaría la programación a través de *sockets*.

En los dos casos anteriores se deben de definir los formatos de mensaje que van a viajar hacia los sistemas, por ejemplo, en qué posición del mensaje va el número de teléfono que se envía, la longitud de los datos, en qué posición del mensaje de retorno va el indicador de aceptación o rechazo, cuáles serán los códigos de error, si cuando la transacción es aceptada se enviará de regreso el nombre del propietario del número telefónico y de qué longitud, en fin, se deben de establecer las reglas en las que se basará el intercambio de la información.

A diferencia de la solución rudimentaria (tipo *trigger file*), implementar una solución a través programación de comunicaciones permitiría que las operaciones, cálculos y validaciones sobre los datos se efectúen en línea y el operador pueda ver inmediatamente los resultados de los mismos. Qué sucede con los casos críticos, como cuando durante un buen período de tiempo se pierde la comunicación con el sistema central, o cuando en el momento de ejecutar una transacción se pierde la comunicación, estos factores son de gran importancia y por lo cual debe de contemplarse al momento del desarrollo de los aplicativos (construcción de los programas).

Además, debe de tomarse en cuenta que una solución utilizando la programación de comunicaciones puede tomar bastante tiempo en ponerla en marcha, y que si se desea hacer una integración con un sistema nuevo, se debe de hacer el ciclo completo para implementarla, teniendo en cuenta que los miembros de la alianza dependerían en gran medida de los conocimientos y habilidades de un reducido número de personas, los expertos en programación de comunicaciones, y eso sería costoso para la integración.

### 3.2.3. Herramientas especializadas

Las herramientas especializadas son un conjunto de productos de *software*, que están orientados a proporcionar los servicios necesarios para poder desarrollar de una forma profesional y eficiente, un proyecto de integración de negocios de cualquier magnitud. Dichas herramientas liberan a los implementadores y usuarios, ya sea programadores, expertos en redes de computadoras, analistas de información, etc., involucrados en la integración, del problema que representa tener que pasar grandes cantidades de horas analizando la forma en que se comunicarán dos arquitecturas diferentes de red, o pasar horas programando la interconexión sobre dos protocolos de comunicación distintos, etc.

Las herramientas están preparadas para trabajar sobre diferentes plataformas de ejecución y diferentes arquitecturas de red a la vez, y también están en la capacidad de proveer formas de definir las reglas en que los diferentes sistemas compartirán e intercambiarán información. Esto no quiere decir que no se requiera de ningún esfuerzo, pero sí se refiere a la mejora de eficiencia, a formas prácticas y fáciles de implementar, a la independencia con respecto a protocolos o arquitecturas de redes de comunicación, que no pueden ofrecer las soluciones rudimentarias o la exhaustiva programación de comunicaciones. Las herramientas especializadas se dividen en tres grupos, herramientas de comunicación (*middleware*), herramientas de programación y herramientas de administración, gracias a sus características. Se habla en detalle de ellas en el capítulo 4, en donde se explica como se les puede identificar, y se hace referencia a algunos productos en particular.

***Middleware***: en la industria de la computación, *middleware* es un término general para cualquier tipo de programación que sirve como ligadura, como intermediario, o como habilitador para que dos programas separados, y usualmente existentes, puedan cooperar. Una aplicación común de *middleware* es el permitir a programas ya escritos, acceder a una base de datos particular, para poder acceder a otras bases de datos. Algunos usuarios probablemente describan el término como programación entre un programa y un *hardware* lógico.

Desde que el término se ha manejado como un término genérico es muy difícil darle una definición concreta. Algunos se refieren al *middleware* como una ligadura, que mantiene juntos todos los componentes en un ambiente de aplicaciones distribuidas. Otros ven el *middleware* como una capa de infraestructura distribuida, la cual mantiene la solución dentro de un rompecabezas de escalabilidad.

Entonces, desde la mayoría de los puntos de vista se está de acuerdo que *middleware* es un elemento clave cuando se enfrentan al desafío de la computación distribuida. La computación es distribuida cuando los programas y los datos con que una computadora trabaja están esparcidos en más de un sistema o red de computadoras.

Para los desarrolladores de aplicaciones enfrentarse con los diferentes protocolos e interfaces, almacenados dentro de los diferentes sistemas y plataformas dentro de un ambiente distribuido podría ser una pesadilla; *middleware* provee una capa de programas que protege a los desarrolladores de aplicaciones de ese problema, proporcionándoles una capa habilitadora de interfaces de aplicación. La figura 1 muestra un esquema de cómo se ubica el *middleware*, dentro de la estructura de un ambiente distribuido, y a su vez muestra la forma en que se estructura el *middleware*.

Figura 1. Estructura y clasificación del middleware

Aplicaciones:	<i>Foreground applications</i>		Programas de cara al usuario	
Proceso distribuido:	ORB/DCOM	Subscripción/Publicación		Data warehouse
Monitor transaccional:	Tuxedo	Encina	MTS	CICS
Modelo distribuido de comunicación	RPC, Modelo cliente - servidor		Mensajería, MQSeries, MSMQ	
Transporte:	<i>LU6.2</i>	<i>TCP/IP</i>	<i>X.25</i>	<i>ISO/LAN</i>

-  Programas de usuario
-  Middleware (Ligadura)
-  Protocolos de transporte

Fuente: [http://www.level8.com/index.cfm?handle=pbg\\_whatismidware](http://www.level8.com/index.cfm?handle=pbg_whatismidware)

Empezando desde la parte inferior de la figura 1, la capa de transporte es la que provee los protocolos de comunicación y acceso a las diferentes arquitecturas de red, LU6.2 para SNA por ejemplo; la capa de aplicación que es la que es vista por los usuarios finales y/o los desarrolladores de aplicaciones que hacen uso del *middleware*, para poder acceder a la capa de transporte olvidándose de todos los problemas que las diferentes configuraciones conllevan.

Dentro del *middleware* existen otras tres capas, las cuales son.

- a. Modelo de comunicaciones distribuidas
- b. Monitores de procesamiento transaccional
- c. Procesamiento distribuido dentro de las cuales existen otras subdivisiones

Los procedimientos remotos RCP, por sus siglas en ingles de *remote procedure calls (RCP-Basse Middleware-DCE)*, es uno de los mecanismos más avanzados para la construcción de aplicaciones distribuidas. Desarrollado para ambientes Unix, los RCPs son ampliamente aceptados como una tecnología capaz de construir aplicaciones distribuidas.

*Middleware* orientado a la mensajería es una clase especial de programas que operan con base en el paso o encolamiento de mensajes. En general, los modelos de aplicaciones distribuidas punto a punto son capaces de soportar interacción asíncrona y/o sincrónica, entre procesos distribuidos que se caracterizan por el manejo de colas de mensajes. En esencia, las colas de mensajes proveen de servicios de alto nivel, soporte para múltiples protocolos, y otros servicios de administración del sistema, creando así una infraestructura que pueda soportar aplicaciones distribuidas en ambientes heterogéneos, que puedan proveer un alto desempeño, escalabilidad y servicios de tiempo real.

**Procesamiento transaccional distribuido:** desde hace mucho tiempo los monitores de procesamiento transaccional han estado reservados para los sistemas *mainframe*, pero desde hace unos 3 ó 5 años, estos monitores han sido habilitados en otros sistemas operativos como *Unix* y *Windows 2000*. Sin los monitores de procesos transaccionales, los clientes, en un modelo cliente servidor, que están acostumbrados a conectarse a un administrador de base de datos, provocarían que dicho administrador iniciara un proceso separado por cada cliente que se conecta; en contraparte, un monitor se convierte en un concentrador de conexiones a la base de datos desde el momento en que los clientes se conectan a él y no al administrado de bases de datos directamente, esto reduce la sobrecarga del sistema y aumenta el rendimiento.

Agentes de solicitud a objetos, ORBs por sus siglas en inglés (*Object Request Brokers*), están dentro de dos categorías *middleware*, los productos que cumplen con los estándares CORBA (*Common Object Request Broker Architecture*) y los que cumplen con la tecnología a COM/DCOM de Microsoft.

CORBA es un conjunto de especificaciones publicadas por el Grupo de Administración de Objetos, una industria cuya misión es la de definir un conjunto de interfaces para la interoperabilidad de los programas orientados a objetos. Estas especificaciones establecen la forma en que los objetos son definidos, creados, procesados, invocados y la forma en que se comunican entre ellos. Un ORB que cumpla con las especificaciones CORBA, puede ser considerado como una pieza del *middleware*, la cual permite que un objeto cliente pueda hacer un requerimiento a un objeto servidor. El equivalente de Microsoft a CORBA, es el *Common Object Model*, y el *Distributed Common Object Model* (COM/DCOM), que soporta la creación y destrucción de objetos, y cuenta con la habilidad de la definición y manejo de objetos, contando con interfaces de requerimientos a objetos.

Por todo lo anterior, se concluye que *middleware* es un conjunto de productos que por sus características son ideales para la implementación de la integración de negocios y además son considerados como herramientas especializadas.



## **4. HERRAMIENTAS ORIENTADAS A LA IMPLEMENTACIÓN DE LA INTEGRACIÓN DE NEGOCIOS**

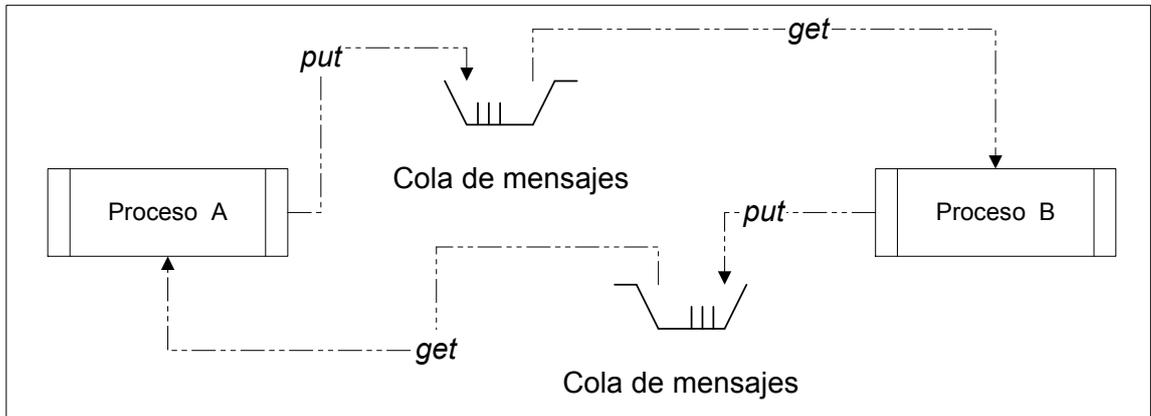
### **4.1. IBM MQSeries**

*MQSeries* de IBM Corporation es una familia de productos que permiten la comunicación dentro de la misma plataforma de sistema operativo o entre plataformas de sistemas operativos distintos. Actualmente existen *MQSeries* para treinta y dos plataformas distintas; entre ellas: *Windows 2000*, *OS/400*, *MVS/ESA*, *AIX*, *OS/2*, *SunOS*, *MVS/ESA*, *Sun Solaris* y *MQSeries Three Tier*.

*MQSeries* crea la arquitectura de comunicación necesaria en un entorno de sistemas abiertos. Los programas se comunican utilizando el API de *MQSeries*, una interfaz de programación de alto nivel y fácil de utilizar que evita a los programadores las complejidades de trabajar con distintos sistemas operativos y redes subyacentes.

El proceso utilizado por el API *MQSeries* para lograr esta comunicación está basado en mensajes y colas de mensajes. Las aplicaciones pueden comunicarse a través de distintas plataformas por medio del intercambio de mensajes o gestión de colas de mensajes proporcionado por *MQSeries*. Los protocolos de comunicaciones se ocultan de las aplicaciones.

Figura 2. **Dos aplicaciones (A y B) utilizando colas de mensajes para intercambiar información**



Las líneas punteadas representan la red de comunicaciones y los canales de comunicación.

Las cinco características más importantes de la gestión de mensajes de *MQSeries* son.

- **La interfaz común de programación de aplicaciones.** Los productos *MQSeries* implementan una interfaz común de programación de aplicaciones, la interfaz de colas de mensajes (MQI), que se utiliza en todas las plataformas en las que se ejecutan las aplicaciones. Las llamadas efectuadas por las aplicaciones y los mensajes que éstas intercambian son comunes. Si se compara con los métodos tradicionales, se ve que se facilita en gran medida la tarea de desarrollar y mantener aplicaciones. También facilita la migración de aplicaciones de una plataforma a otra.

- **Entrega asegurada de los mensajes.** *MQSeries* puede transferir los datos asegurando la entrega de éstos, incluso si ocurre una falla en el sistema *MQSeries* garantiza que el mensaje enviado será entregado cuando el sistema se recupere. Algo tan importante como la entrega asegurada de mensajes es la garantía de que los mensajes no se duplicarán.
- **Aplicaciones independientes del tiempo.** Las aplicaciones que se están comunicando no tienen que estar activas al mismo tiempo. Con la gestión de colas de mensajes, el intercambio de mensajes entre programas emisores y receptores es independiente del tiempo. Esto significa que las aplicaciones emisora y receptora se desacoplan de manera que el emisor pueda continuar el proceso sin tener que esperar a que el receptor acuse recibo del mensaje. De hecho, la aplicación de destino ni siquiera tiene que estar ejecutándose cuando se envía el mensaje sino que puede recuperarlo después de arrancar.
- **Aplicaciones en paralelo.** El proceso de entrega de los mensajes depende del diseño de las aplicaciones. Una aplicación puede dividirse en módulos funcionales que se comunican entre ellos por medio de mensajes. De esta manera, los módulos ejecutados en sistemas diferentes pueden recibir los mensajes en tiempos distintos o al mismo tiempo (en paralelo).
- **El desarrollo de aplicaciones es más rápido.** Las complejidades de las redes de comunicación (protocolos, recuperación en caídas de la línea de comunicación, sincronía, etc.) se ocultan de los desarrolladores de aplicaciones porque son manejadas por *MQSeries*. Esto permite que el desarrollo de aplicaciones se realice más rápidamente.

## **4.1.1. Mensajes y colas**

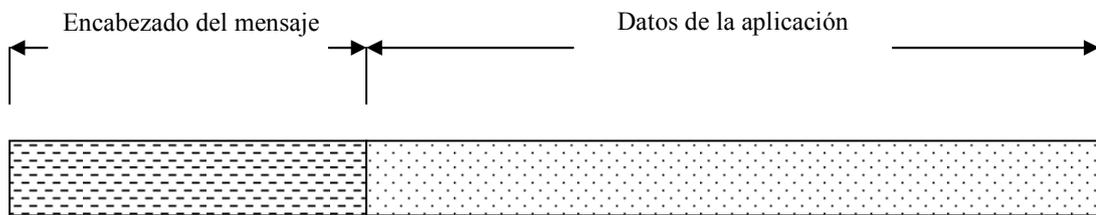
### **4.1.1.1. ¿Qué es un mensaje?**

Un mensaje es una unidad de información que tiene un significado para aquellas aplicaciones que lo utilizan. Podría ser, por ejemplo: un requerimiento de un servicio, una respuesta, un reporte, un error, un anuncio o cualquier información que una aplicación desee comunicarle a otra.

En *MQSeries*, los mensajes se componen de dos partes, un encabezado de mensaje y los datos de la aplicación. Los programas de aplicación que utilizan los datos de la aplicación definen el contenido y la estructura de los mismos. El encabezado de mensaje identifica el mensaje y contiene otros atributos o información de control como la fecha y la hora en que se creó el mensaje, el tipo de mensaje y la prioridad que ha asignado al mensaje la aplicación emisora.

La figura 3 representa un mensaje y muestra la manera lógica en que éste se divide en datos del mensaje y datos de la aplicación.

Figura 3. **Representación de un mensaje**



**El descriptor (o encabezado) del mensaje y los datos de la aplicación aparecen como partes individuales. La información específica a la aplicación es la parte del mensaje que forma los datos de la aplicación**

#### **4.1.1.2. Permanencia y prioridad de los mensajes**

Los mensajes tienen dos atributos importantes definidos en el encabezado del mensaje: permanencia y prioridad.

Un mensaje recibe el nombre de permanente si perdura cuando *MQSeries* se reinicia. Esto implica que el mensaje se debe anotar, o que se debe salvar, y que se puede volver a crear como parte del procedimiento de recuperación.

Cada mensaje *MQSeries* tiene una prioridad asignada por la aplicación emisora. La prioridad, que es un número del 0 al 9, puede afectar al orden en el que se recupera un mensaje de una cola y también a la forma en que se generan los sucesos de activación (*triggering*).

El mecanismo de activación es un recurso de que disponen algunos productos *MQSeries* que permiten arrancar una aplicación automáticamente cuando se cumplen las condiciones predefinidas en una cola. Dichas condiciones pueden ser: la recepción de cualquier mensaje de los mensajes que estén por encima de una prioridad determinada, el número de mensajes de una cola, etcétera.

### **4.1.1.3. Tamaños de los mensajes**

El tamaño de los mensajes que un programa de la aplicación puede colocar en una cola está limitado por.

- La longitud máxima de mensajes definida para la cola receptora
- La longitud máxima de mensajes definida para el gestor de colas
- La longitud máxima de mensajes aceptada por la plataforma

Si el tamaño de los datos que un programa de aplicación necesita colocar en una cola sobrepasa el límite, el programa deberá dividir los datos en un número determinado de partes y colocar cada una de estas partes en la cola como si fueran mensajes independientes.

### **4.1.1.4. ¿Qué es una cola?**

Una cola es un tipo de lista que se utiliza para almacenar mensajes hasta que una aplicación los recupere. Las colas existen independientemente de las aplicaciones que las utilizan. Una cola puede estar.

- En el almacenamiento principal, si es temporal.
- En un disco o almacenamiento auxiliar similar, si debe conservarse en caso de que fuera necesario su recuperación
- En ambos lugares, si se está utilizando y también debe conservarse por si fuera necesario su recuperación

En *MQSeries* cada cola pertenece a un gestor de colas que es el responsable de mantenerlas. El gestor de colas transfiere los mensajes que recibe a la cola adecuada. Las colas pueden estar en el sistema local, en cuyo caso se denominan colas locales o en otro gestor de colas, en cuyo caso se denominan colas remotas.

#### **4.1.1.5. Algoritmos de recuperación de mensajes**

Las aplicaciones con autorización adecuada pueden recuperar mensajes de una cola según ciertos algoritmos de recuperación. Estos algoritmos definen la manera en que los mensajes son sacados y puestos en las colas.

- Primero en entrar, primero en salir (FIFO).
- Prioridad del mensaje, tal como está definida en el encabezado del mensaje. Los mensajes que tienen la misma prioridad se recuperan por orden FIFO.
- Una petición de programa para un mensaje determinado.

#### **4.1.1.6. Atributos de la cola**

Los atributos de la cola determinan qué ocurre cuando las aplicaciones hacen referencia a la cola en las llamadas de la interfase de programación. Algunos de los atributos de las colas son.

- Obtención (*get*) habilitada: las aplicaciones pueden recuperar mensajes de la cola.
- Poner en cola (*put*) habilitada: las aplicaciones pueden poner mensajes en la cola.
- Exclusividad: si el acceso a la cola es exclusivo para una aplicación o si las aplicaciones la comparten.
- Profundidad de la cola: el número máximo de mensajes que se pueden almacenar en la cola simultáneamente.
- El tamaño máximo de mensajes que se pueden poner en la cola.

#### 4.1.1.7. Tipos de colas

En *MQSeries*, hay varios tipos de colas. Cada tipo de cola puede ser manipulado con comandos de *MQSeries* y está asociado a las colas de formas diferentes:

Una cola local pertenece al gestor de colas al que está conectada la aplicación.

Una cola remota identifica una cola que pertenece a otro gestor de colas. A la cola remota normalmente se le da una definición local. La definición especifica el nombre del gestor de colas remoto en que se encuentra la cola así como el nombre de la propia cola. La información que se especifica al definir una cola remota permite al gestor de colas encontrar el gestor de colas remoto para que todos los mensajes destinados a esta cola remota vayan al gestor de colas correcto.

Una cola alias permite que las aplicaciones accedan a las colas haciendo referencia a las mismas indirectamente. Cuando se utiliza un nombre de cola alias en una llamada a la MQI, el nombre se convertirá en el de una cola de mensajes en el momento de la ejecución. De esta forma se le permite cambiar las colas que utilizan las aplicaciones sin cambiar en ningún aspecto la aplicación; simplemente se cambia la definición del alias.

Una cola modelo define un conjunto de atributos de cola que se utilizan como plantilla para una cola dinámica. Las colas dinámicas las crea el gestor de colas cuando una aplicación efectúa una petición de abrir una cola especificando la cola que es una cola modelo. La cola dinámica que se crea de esta manera es una cola local cuyo nombre está especificado por la aplicación y cuyos atributos son los de la cola modelo. Las colas dinámicas pueden ser de dos tipos.

**Temporales**, este tipo de cola se suprime cuando se cierra la cola y no sobrevive cuando se arranca un gestor de colas. Las colas pueden utilizarse para almacenar sólo mensajes no permanentes.

**Permanentes**, este tipo de cola no se suprime cuando se cierra la cola, a menos que el programa de aplicación solicite explícitamente que se suprima. La cola perdura a un reinicio de un gestor de colas, y se puede utilizar para almacenar mensajes permanentes y no permanentes.

**Nota:** algunos productos *MQSeries* no dan soporte a todos los tipos de objetos de cola relacionados anteriormente.

#### **4.1.2. Gestor de colas (*Queue Manager*)**

Es el componente de *MQSeries* encargado de manejar las colas. Provee una interfase de programación (*Message Queue Interface, MQI*) que permite a las aplicaciones acceder las colas y los mensajes que ellas contienen. Esta interfase de programación es simple y consistente en todas las plataformas soportadas por *MQSeries* y, además, protege efectivamente a las aplicaciones de tener que saber como son manejadas físicamente las colas y los mensajes por el gestor de colas.

En algunos entornos, por ejemplo con *MQSeries* para OS/400, sólo se puede utilizar un gestor de colas a la vez en una máquina; en otros entornos, por ejemplo con *MQSeries* para MVS/ESA, se puede ejecutar a la vez más de un gestor de colas en una máquina. Típicamente no es necesario crear más de un gestor de colas.

Las aplicaciones realizan las llamadas a la interfase de programación (MQI), estas llamadas son interpretadas y ejecutadas por el gestor de colas. Para los mensajes entrantes, el gestor de colas los dirige a sus colas de destino respectivas; para los mensajes salientes el gestor de colas envía el mensaje al gestor de cola de destino. El gestor de colas garantiza que el mensaje se coloca en la cola correcta.

El primer objeto que debe crearse después de la instalación de *MQSeries* es el gestor de colas (*Queue Manager*). Luego podrán crearse otros objetos como colas, canales y procesos que pertenecerán al gestor de colas que los haya creado.

#### **4.1.2.1. Canal de mensajes**

Es un enlace de una sola vía entre dos gestores de colas para la transmisión de mensajes. Cada extremo del canal de mensajes tiene una definición que debe definirse por aparte. Las dos definiciones contienen el nombre del canal. Entre las características de estas definiciones está aquella que identifica el extremo que enviará mensajes y el extremo que los recibirá. Se define también el protocolo de comunicaciones que se utilizará. Un canal de mensajes está compuesto por.

- Una cola de transmisión localizada en el extremo que envía. Sólo la definición del canal en el extremo que envía contiene el nombre de la cola de transmisión. Esta es realmente una cola local, pero tiene un atributo cuyo valor indica el papel especial de esta.
- Un agente de canales de mensajes en cada gestor de colas. Es un programa que controla las comunicaciones y que forma parte del gestor de colas.

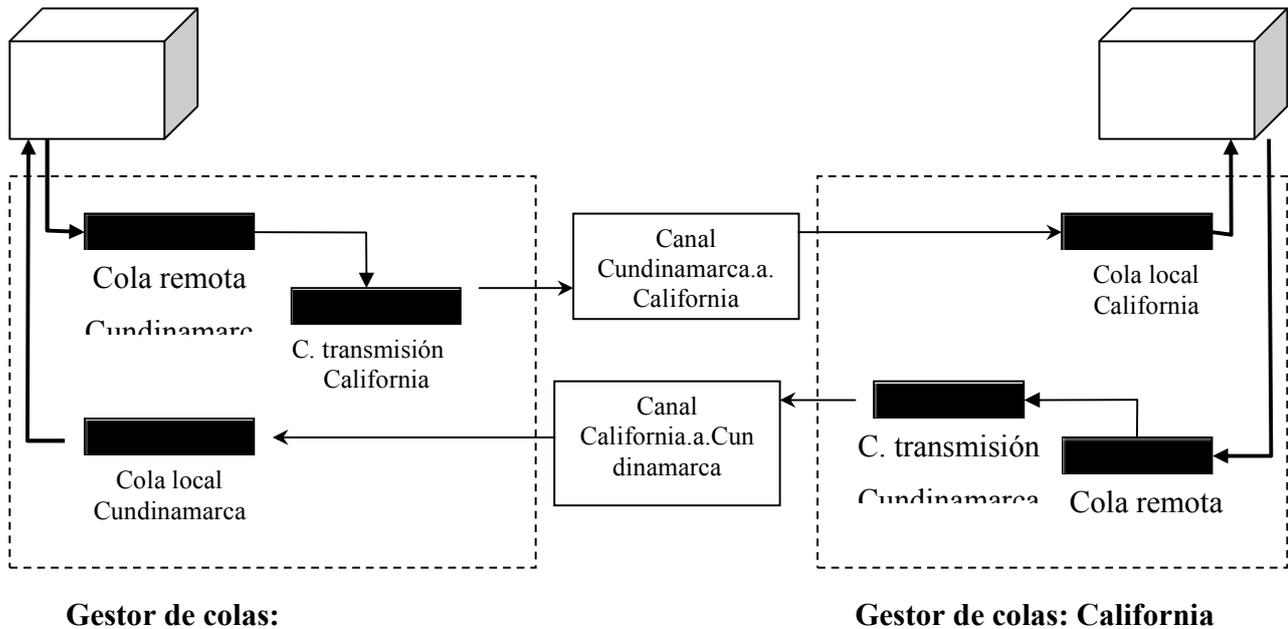
- Un enlace de comunicaciones que puede ser una conversación SNA o una conexión TCP/IP.

#### 4.1.2.2. Procesos

Los sucesos de activación (*triggering*) permiten arrancar una aplicación cuando se producen determinados eventos o condiciones. Los procesos se utilizan para indicarle a un monitor de sucesos de activación el nombre de la aplicación que se debe arrancar y algunas características con las que debe hacerse.

En la figura 4 se muestra una configuración sencilla de comunicación entre dos gestores de colas en plataformas distintas. Las aplicaciones A y B mantienen una conversación de la siguiente forma: cuando A quiere enviar un mensaje a B, coloca dicho mensaje en la cola remota definida en el gestor de colas Cundinamarca, el gestor de colas se encarga de poner el mensaje en la cola de transmisión y de allí por medio del canal respectivo, el mensaje es llevado a la cola local definida en el gestor de colas California, la aplicación B, envía mensajes a la aplicación A de manera similar. Ambas aplicaciones obtienen los mensajes de sus colas locales. Los nombres de los canales indican claramente la dirección (vía) que llevan los mensajes que pasan por ellos. Únicamente son necesarios dos canales para comunicar recíprocamente a dos gestores de colas. El gestor de colas Cundinamarca sabe que los mensajes enviados al gestor de colas California deben colocarse en la cola remota del mismo nombre. Esto quiere decir que un gestor puede comunicarse con más de un sistema remoto.

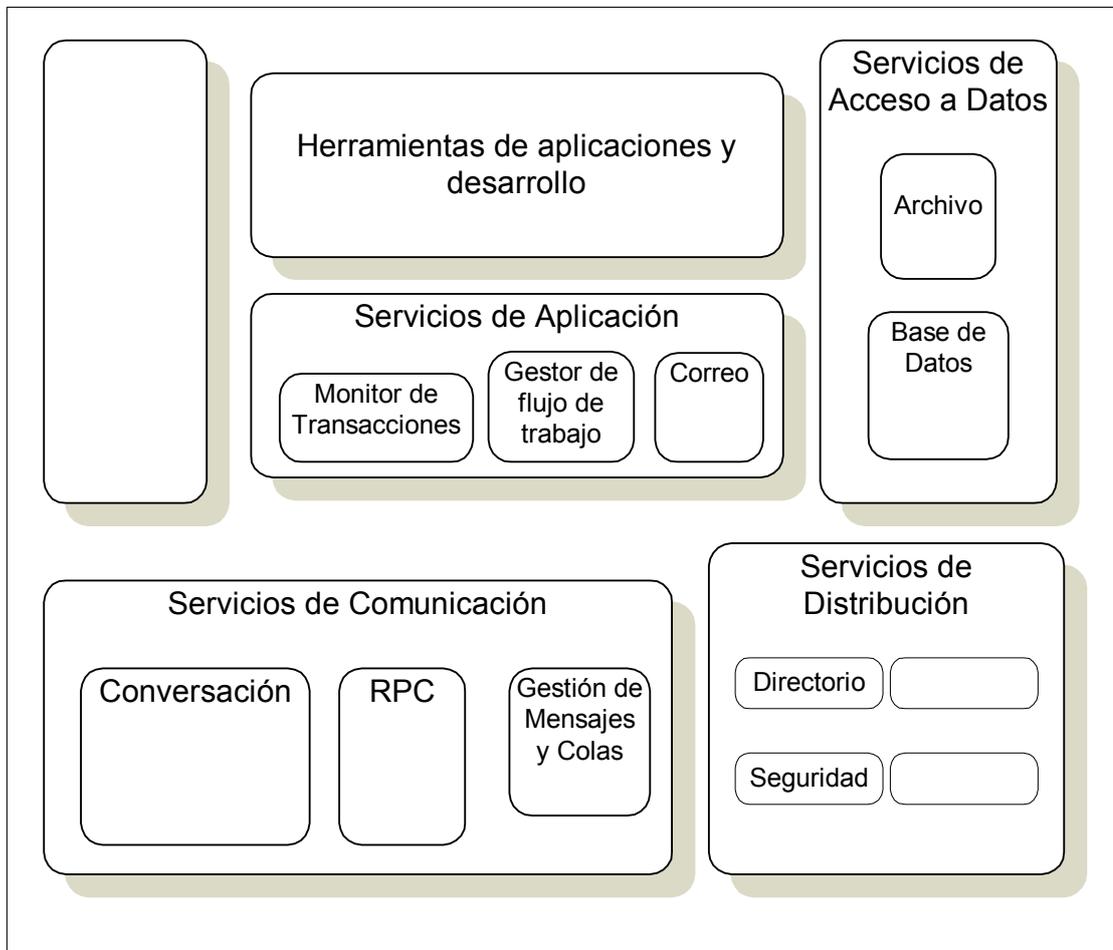
Figura 4. Esquema simple de comunicación entre dos gestores de colas que podrían estar en plataformas distintas



### 4.1.3. Seguridad

Para comprender las funciones de seguridad proporcionadas por *MQSeries* es necesario conocer la posición lógica de *MQSeries* en relación con otros componentes de un sistema y los servicios de seguridad que se exponen. La posición relativa de *MQSeries* respecto a otros componentes de un sistema se ilustra en la figura 5, donde *MQSeries* está representado mediante el componente de envío de mensajes y gestión de colas.

Figura 5. Componentes de un sistema



En la figura 5 se muestra que los servicios de seguridad están separados de otros componentes, tales como los servicios de aplicación, los gestores de recursos y las comunicaciones. Esta separación significa que sólo es necesario que un componente del nodo proporcione los servicios de seguridad y que los otros componentes simplemente llaman a los servicios apropiados cuando se necesitan.

Los servicios de seguridad que puede proporcionar el componente de seguridad (separado) son.

- Identificación y autenticación (*I&A*).

Este servicio forma la base de muchos de los otros servicios e incluye el suministro de un identificador de usuario y la verificación de que el identificador es válido (es decir, que representa al usuario real y no a un intruso que finge ser un usuario válido).

- Autorización

Es el control de acceso y depende de la disponibilidad de un identificador de usuario para compararlo con las listas de control de acceso.

- Confidencialidad de datos o codificación.

- Integridad de los datos.

Aunque los datos puedan verse (es decir, no estén cifrados), el servicio de integridad de datos asegura que no se modifiquen los datos.

- No rechazo.

Este servicio proporciona alguna forma de señal (como por ejemplo una firma digital) que garantiza que una parte determinada de datos se ha originado en un usuario determinado.

#### 4.1.4. Conexión de aplicaciones

Antes de que una aplicación se conecte a un gestor de colas tendrá que someterse a alguna forma de procedimiento I&A. Éste puede consistir en tener que proporcionar un identificador de usuario y una contraseña o puede ser algún proceso más elaborado (como por ejemplo la identificación por tarjeta inteligente).

Como alternativa es posible que no haya ningún requisito para el procedimiento I&A y, por lo tanto, ningún identificador de usuario asociado a la aplicación (por ejemplo, el edificio en el que está el sistema puede estar bien protegidos y no considerarse necesarios los identificadores de usuario).

La consecuencia de ello es que cada aplicación que emite llamadas a la interfase de programación de *MQSeries* (MQI) tiene un identificador de usuario asociado (que puede ser nulo) que se utiliza para autorizar el uso de determinadas funciones y opciones en objetos determinados (generalmente colas). Esto significa que cualquier identificador de usuario de aplicación que intente acceder a los recursos de *MQSeries* debe tener la autorización adecuada.

Puesto que el procedimiento I&A tiene lugar antes de que la aplicación se conecte al gestor de colas, deberán ser otros componentes, no *MQSeries*, los que proporcionen el servicio I&A. *MQSeries* sólo es responsable de capturar el identificador de usuario para utilizarlo para proporcionar otros servicios de seguridad, como por ejemplo la autorización. (El identificador de usuario se capta cuando la aplicación se conecta al gestor de colas.)

Son excepciones de este ámbito de responsabilidad las plataformas OS/2 Warp y *Windows 2000*, donde *MQSeries* ofrece un servicio para proporcionar un identificador de usuario para las aplicaciones cuando éstas se conectan al gestor de colas.

De modo similar, generalmente es responsabilidad de algún otro componente proporcionar el servicio de autorizaciones, teniendo *MQSeries* la responsabilidad de llamar a dicho servicio. Esto funciona satisfactoriamente para sistemas tales como MVS y AS/400 donde existe una interfaz estandarizada al servicio de autorizaciones.

Sin embargo, para las plataformas *OS/2 Warp*, *Windows 2000* y *Unix*, no se proporciona ningún servicio de autorizaciones ni interfaz estándar y por consiguiente *MQSeries* proporciona una interfaz propia.

#### **4.1.4.1. Mensajes *MQSeries***

La función básica de *MQSeries* consiste en pasar mensajes entre aplicaciones. El encabezado del mensaje contiene un campo (llamado identificador de usuario) donde se puede poner un identificador de usuario para que la aplicación que obtiene el mensaje sepa en qué usuario se ha originado el mensaje.

Por consiguiente, este aspecto del funcionamiento de *MQSeries* proporciona un servicio de identificación (asociando un identificador de usuario con el mensaje); *no* proporciona un servicio de autenticación. Actualmente es responsabilidad de las aplicaciones *MQSeries* proporcionar alguna señal de autenticación necesaria y verificar dicha señal.

#### **4.1.4.2. Seguridad de punto a punto**

Además de proporcionar servicios como gestor de recursos local, un gestor de colas también proporciona gestión de colas distribuidas, permitiendo que los mensajes se distribuyan por la red de gestores de colas. Esta función de mensajes distribuidos se proporciona por medio de canales *MQSeries*.

Cada canal se compone de un par de programas agente de canal de mensajes (MCA), que proporcionan el protocolo para realizar una sola entrega segura de mensajes utilizando un mecanismo de transporte implícito para intercambiar mensajes.

Cuando dos MCA establecen comunicación, puede que sea necesario que cada uno de ellos verifique la identidad del otro. Esto sería así en el caso de que un gestor de colas no confiara en la conexión o la identidad del gestor de colas asociado (por ejemplo, si fueran propiedad de empresas independientes).

#### **4.1.5. Recuperación**

##### **4.1.5.1. Unidades de recuperación**

En un único gestor de colas, una unidad de recuperación es una parte del trabajo que cambia datos de un punto de coherencia a otro. Un punto de coherencia (también llamado punto de sincronismo o punto de compromiso) es un momento en el que todos los datos recuperables a los que accede el programa de aplicación son coherentes. Esto significa que una unidad de recuperación comienza con un intento de cambiar datos y termina en un punto de coherencia.

Por ejemplo, es posible que una transacción bancaria transfiera fondos de la cuenta A a la cuenta B. En primer lugar, un programa de débito sustrae la cantidad de la cuenta A. Posteriormente, debe ingresarse la cantidad en la cuenta B. Después de restar la cantidad de la cuenta A, las dos cuentas son incoherentes y el gestor de colas no puede comprometerse. El paso siguiente del proceso es para que el programa de cargar en cuenta transfiera un mensaje a una cola para el programa de crédito, proporcionando información sobre la cantidad de dinero y la cuenta que debe saldarse.

Cuando el programa de crédito recibe el mensaje de la cola, lleva a cabo el abono en la cuenta B y notifica al programa de débito que ya ha realizado la tarea. Ahora los datos son coherentes y la aplicación puede anunciar un punto de coherencia y mostrar los cambios a las demás aplicaciones.

Si se produce un error en una unidad de recuperación, se puede utilizar el gestor de colas para eliminar cambios de los datos, devolviendo los datos a su estado del inicio de la unidad de recuperación; es decir, el gestor de colas deshace el trabajo.

#### **4.1.5.2. Mantener la coherencia de los datos**

Si los datos que gestiona un gestor de colas deben ser coherentes con los datos de otros subsistemas, todos los datos que se modifiquen en uno deben reflejarse en un cambio en los demás. Antes de que un sistema comprometa los datos modificados, debe saber que el otro sistema (o los otros sistemas) puede efectuar los cambios correspondientes. Por este motivo, los sistemas deben comunicarse entre sí.

Durante un compromiso de dos fases, un subsistema coordina el proceso. Dicho subsistema recibe el nombre de coordinador; los otros son los participantes. En primer lugar, el coordinador comprueba que todos los participantes estén preparados para llevar a cabo el compromiso; que es la fase uno. Cuando el coordinador recibe las respuestas positivas de los participantes que están preparados para el compromiso, emite la petición de compromiso; que es la fase dos. Ahora los participantes, junto con el coordinador, se comprometen con los datos modificados. Todos los datos vuelven a ser coherentes.

Durante un compromiso de fase única no hay coordinador como en las interacciones. Cuando se emite la llamada de compromiso, la respuesta a la llamada es que:

- el compromiso se realizó satisfactoriamente
- la llamada se realizó parcialmente
- se produjo una anomalía en la llamada.

La aplicación que emite la llamada decidirá qué acción llevar a cabo, si es necesaria alguna.

Al reiniciar un gestor de colas después de una interrupción, éste debe determinar si debe comprometer o restituir las unidades de recuperación que estaban activas en el momento del error. Para algunas unidades de recuperación el gestor de colas dispone de suficiente información para tomar decisiones. No en cambio para otras y deberá obtener la información del coordinador cuando se restablezca la conexión.

La acción de punto de sincronismo no puede producirse directamente en enlaces remotos. En su lugar, se utiliza la característica de entrega de mensajes asegurada de los productos *MQSeries*. Una aplicación envía una petición de compromiso en forma de mensaje a un sistema remoto con la seguridad de que se entregará dicho mensaje. Los diseñadores de la aplicación deben asegurarse de haber creado las peticiones de compromiso y de haberlas respondido. También deben tener en cuenta situaciones en que el compromiso no pueda aceptarse y responder a la aplicación solicitante del compromiso apropiadamente.

Un ejemplo de problemas de compromisos remotos es cuando el sistema remoto no está disponible. Los mensajes para dicho sistema permanecerán en las colas de transmisión hasta que se puedan entregar.

Cuando por fin se realice la entrega, se podrán efectuar todos los compromisos que se desee. En tal situación se trataría de aplicaciones independientes en el tiempo. Serán los diseñadores de aplicaciones los que deben considerar cómo y cuándo necesitan que los datos compartidos entre sus sistemas remotos sean coherentes.

#### **4.1.6. Administración**

La administración de una aplicación que utiliza productos *MQSeries* la realiza un administrador del sistema, un programador del sistema o un operario del sistema que disponga de la autorización adecuada. El administrador es el responsable de supervisar los productos *MQSeries* y los recursos que están utilizando, así como realizar los cambios que sean necesarios para que las aplicaciones sigan funcionando sin problemas.

*MQSeries* suministra un conjunto de comandos, interfaces de comandos y programas de utilidad que proporcionan funciones de administración de mensajes y colas, incluyendo por ejemplo.

- Administración de colas
- Administración de canales
- Administración de seguridad
- Crear y suprimir colas
- Detener o iniciar canales
- Cambiar la autorización de acceso para las colas

Estas funciones las proporcionan los comandos de *MQSeries* y el gestor de colas las procesa de las formas siguientes.

- **Ingresados por el administrador del sistema.** Un administrador del sistema entra un comando y el gestor de colas lo procesa inmediatamente. Se da una respuesta al administrador. El método que se utiliza para entrar comandos depende de la plataforma y es el método habitual que se utiliza para los comandos de entrada en la plataforma. Podría ser una consola de comandos, una línea de comandos, un recurso de panel de visualización o algunas otras técnicas.
- **Almacenados como lista de comandos.** Los comandos se almacenan como una lista en un archivo. Posteriormente, el administrador puede hacer que el gestor de colas lea los comandos del archivo, los procese y genere respuestas.
- **Almacenados como mensajes en una cola.** Un programa de aplicación genera un comando y lo transfiere a una cola de comandos del gestor de colas. El gestor de colas extrae el mensaje de la cola, procesa el comando incluido en el mensaje y genera una respuesta.

Los productos *MQSeries* proporcionan los tipos de comandos siguientes:

- comandos (MQSC) de *MQSeries* – legibles,
- comandos de formato de comando programable (PCF) - sólo legibles por la máquina.

Los dos conjuntos de comandos llevan a cabo funciones parecidas, pero su finalidad es utilizarlas en distintas situaciones.

#### **4.1.6.1. Comandos MQSC**

Los comandos *MQSeries* (MQSC) proporcionan un método uniforme de emitir comandos en un formato que pueda leerse por todas las plataformas de *MQSeries*. Se

utilizan en aquellas situaciones en que las personas puedan leer los comandos, por ejemplo, siempre que se entren los comandos mediante una línea de comandos. Las respuestas a estos comandos también puede leerse, sin embargo el contenido y el formato depende de la plataforma en la que se utilizan.

#### **4.1.6.2. Comandos en formato de comando programable**

Los comandos en *formato de comando programable* (PCF) de *MQSeries* están concebidos para que los usen los programas de aplicación que proporcionan recursos para la administración de gestores de colas y los recursos que están utilizando (por ejemplo, colas y canales), a partir de un solo punto de la red. El formato de estos comandos y de sus respuestas es independiente del entorno en el que se utilizan.

#### **4.1.6.3. Colas de comandos**

Algunos productos *MQSeries* disponen de una cola de comandos que acepta comandos que contienen mensajes de administración. Estos mensajes reciben el nombre de *mensajes de comandos*. El gestor de colas tiene un *servidor de comandos* que procesa los mensajes de comandos de la cola de comandos.

Los mensajes de comandos se pueden colocar en la cola mediante una aplicación local o remota. Los mensajes de comandos los procesa el servidor de comandos y devuelve una respuesta al programa de aplicación utilizando el nombre de la cola de respuestas que se encuentra en el mensaje de comandos original.

#### **4.1.6.4. Programas de aplicación de administración**

Los productos *MQSeries* proporcionan programas de aplicación, denominados programas de utilidad de administración, que permiten que el administrador del sistema

de un procesador controle el producto *MQSeries* que se está ejecutando en dicho procesador.

Algunos de estos programas de utilidad también permiten al administrador controlar los productos *MQSeries* que se están ejecutando en otros procesadores, y por lo tanto, se denomina administración remota.

Es aquellas situaciones en que no haya disponible ningún programa de utilidad de administración, puede crear su propio programa de aplicación que realice estas funciones. Esta aplicación puede gestionar cualquier gestor de colas (local o remoto) que tenga una cola de comandos, transfiriendo a dicha cola mensajes de comandos. Cuando se procesa un comando se devuelve un mensaje de respuesta a la cola de respuestas que el programa de aplicación haya especificado en el mensaje de comandos original.

#### **4.1.7. Ambiente cliente servidor**

Un *cliente MQSeries* es una parte de un producto *MQSeries* que se puede instalar en una máquina sin tener que instalar todo el gestor de colas. Acepta llamadas de la interfaz de programación (MQI) que provienen de los programas de aplicación, y pasa las peticiones de la MQI a un *servidor MQSeries* que se esté ejecutando en otro procesador.

El *servidor MQSeries* es un gestor de colas completo que puede aceptar llamadas de la MQI que provienen directamente de los programas de aplicación que se están ejecutando en el procesador del servidor; además, puede aceptar peticiones de la MQI que procedan de clientes *MQSeries*.

Esto permite que el usuario utilice una aplicación que utilice la MQI que se está ejecutando en una máquina, la máquina cliente y el propio gestor de colas que se está ejecutando en otra máquina.

Los clientes y los servidores pueden ser útiles en determinadas situaciones.

- Cuando no haya implementación completa de *MQSeries* en la máquina (por ejemplo, porque es una plataforma de D.O.S.)
- Cuando la máquina cliente es demasiado pequeña, o si la capacidad de proceso es insuficiente, para ejecutar un gestor de colas completo con un buen rendimiento
- Cuando desee permitir que el programa de aplicación del procesador cliente se conecte a varios gestores de colas situados en distintos procesadores de servidor
- Cuando desee reducir el esfuerzo de la administración de sistemas

El computador puede ejecutar una aplicación *MQSeries* en un entorno *MQSeries* completo y en un entorno *MQSeries* cliente sin cambiar el código. Sin embargo, las bibliotecas que *MQSeries* utilice durante la edición de enlaces determinan el entorno en el que debe ejecutarse la aplicación.

Cuando un programa de aplicación en el cliente emite una llamada de la MQI, el cliente dará formato a los valores de los parámetros de la llamada en una petición de la MQI y envía la petición al servidor. El servidor recibe la petición, lleva a cabo la acción especificada en la petición y devuelve una respuesta al cliente. La respuesta la utiliza el cliente para generar información que se devuelve al programa de aplicación utilizando el mecanismo normal de devolución de la MQI.

Otra función soportada por los clientes *MQSeries* es la capacidad de que un programa de aplicación se conecte a más de un gestor de colas a la vez, estando los gestores de colas en distintos procesadores o en el mismo procesador.

#### 4.1.7.1. Comunicación entre clientes y servidores

Un cliente *MQSeries* se comunica con un servidor *MQSeries*, utilizando un canal de la MQI, que se utiliza para transferir peticiones de llamadas de la MQI desde el cliente hasta el servidor y devolver respuestas desde el servidor hasta el cliente.

Los canales de la MQI difieren de los canales de mensajes (que se utilizan para conectar gestores de colas) en dos sentidos:

- El canal de la MQI es bidireccional. Un canal de la MQI se puede utilizar para enviar peticiones en una dirección y respuestas en la dirección contraria.

Con canales de mensajes, los datos sólo se pueden pasar en una dirección. Si se necesita la comunicación bidireccional entre dos gestores de colas (por ejemplo, en la situación en que los mensajes de respuesta se tuvieran que enviar al mismo gestor de colas que manejó un mensaje de petición inicial) entonces, se necesitarían dos canales de mensajes: uno para manejar mensajes que van en una dirección y otro para los mensajes que van por la otra dirección.

- La comunicación en un canal de la MQI es sincrónica. Cuando se transmite una petición de la MQI desde un cliente a un servidor, el producto cliente *MQSeries* debe esperar una respuesta del servidor antes de que pueda enviar la siguiente petición de la MQI.

Con canales de mensajes, el tráfico de mensajes en el canal es independiente en el tiempo. Se pueden enviar varios mensajes desde un gestor de colas a otro sin que el gestor de colas emisor tenga que esperar una respuesta del gestor de colas receptor.

El protocolo de transmisión que se ha de utilizar en un canal de la MQI se especifica como parte de la definición del canal. El programa de aplicación de *MQSeries* ignora el protocolo concreto que se está utilizando en el canal. Además, en la situación en que un programa de aplicación está conectado a más de un servidor *MQSeries*, los canales de la MQI se utilizan para que esas conexiones puedan utilizar distintos protocolos. Por ejemplo, un programa de aplicación podría conectarse con un gestor de colas que utilice TCP/IP en un canal y con otro gestor de colas que utilice NetBIOS en un canal distinto.

Tanto con los canales de la MQI como con los canales de mensajes, se necesitará una definición de canal en el extremo de cada canal y cada una de estas definiciones deberá incluir un tipo de canal y un nombre de canal. Puede elegir utilizar diferentes tipos de canal según la aplicación que esté diseñando, pero debe utilizar el mismo nombre de canal en ambos extremos del canal.

## **4.2. *MQSeries Integrator***

La primera versión de IBM *MQSeries Integrator* ofrece a las organizaciones la habilidad de conectar e integrar aplicaciones residentes en múltiples plataformas en diferentes áreas geográficas. Adicionalmente, los mensajes en *MQSeries Integrator* versión 1.0 pueden ser descritos, transformados por un proceso de formateo, y dirigidos a su destino utilizando reglas simples.

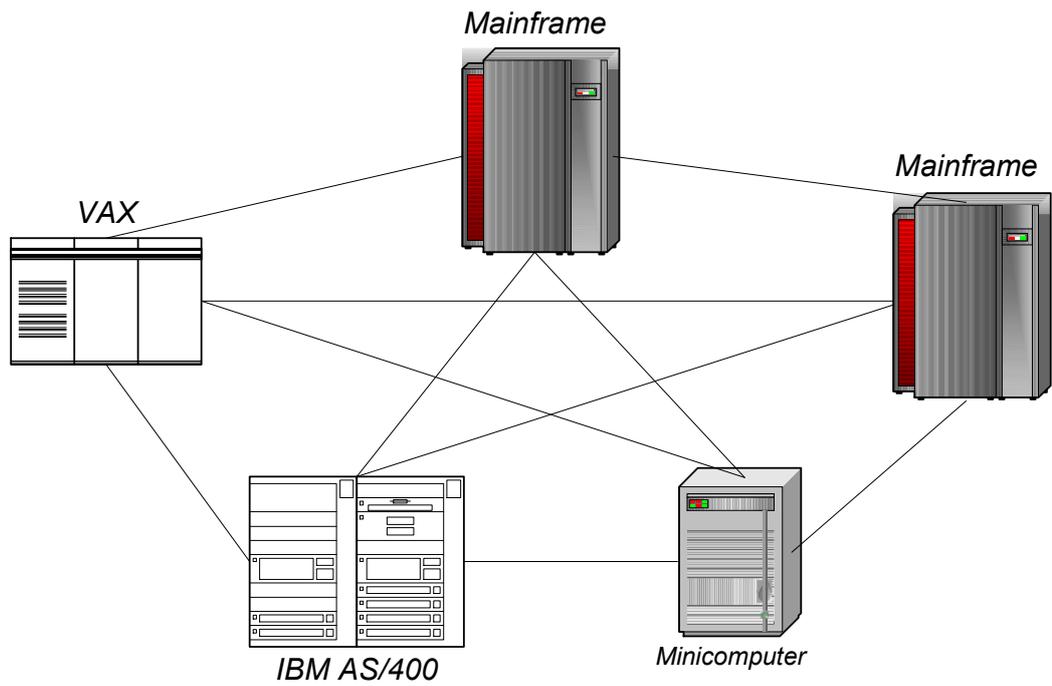
#### 4.2.1. Agentes de mensajes (*Message brokers*)

El intercambio **asincrónico** de mensajes se ha convertido en el estándar *de facto* para el intercambio de datos. No importa si es negocio a negocio, negocio a cliente o intercomunicación entre aplicaciones dentro de la misma organización, la mensajería se ha convertido en la solución a escoger. Al principio, los agentes de mensajes fueron creados para facilitar la complejidad de grandes ambientes de mensajería y para simplificar las necesidades de conexiones a las redes de trabajo.

Por ejemplo: en un sistema construido con *MQSeries* que involucre cinco diferentes gestores de colas, para que cada gestor de colas tenga acceso a los demás se necesitarían diez pares de canales para enviar y recibir mensajes; estas son veinte definiciones de canales. En la figura 6 cada línea representa dos pares de canales (enviador/receptor).

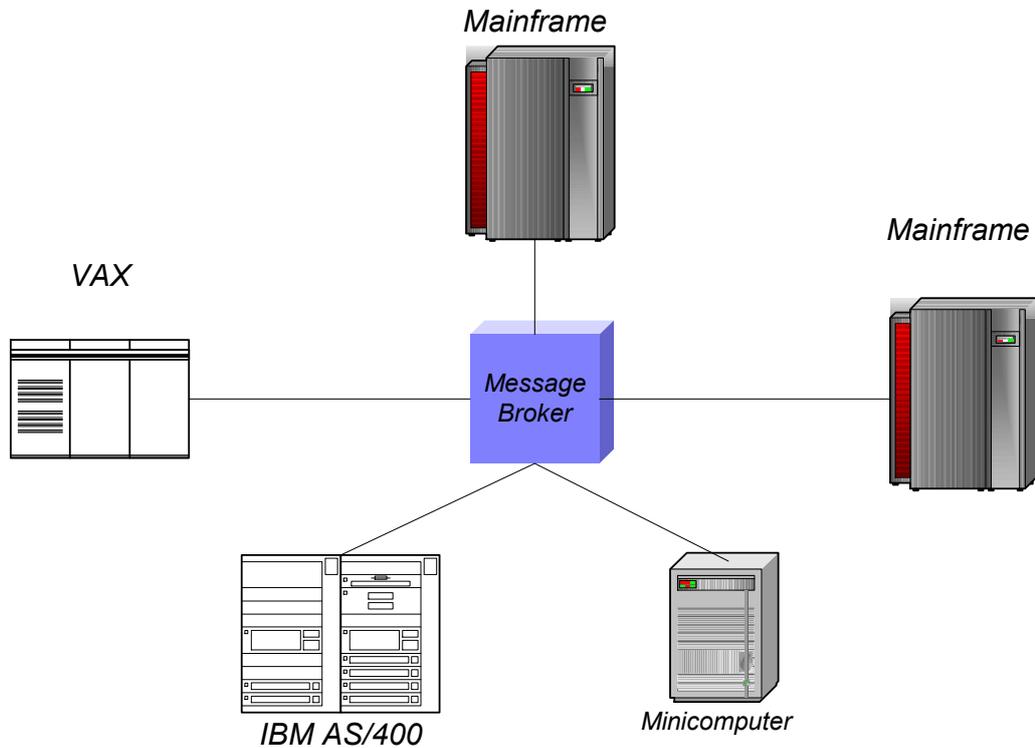
La siguiente fórmula:  $n * (n - 1) = c$ , donde  $n$  es el número total de gestores de colas y  $c$  es igual al número de canales que se necesitan. Este tipo de configuración es comúnmente llamada estrella. Si se desea añadir un nuevo gestor de colas al ambiente, un administrador de *MQSeries* necesitaría definir  $(nw - 1) * 2$  canales nuevos, en donde  $nw$  es el número de nuevos gestores de colas. En el ejemplo, sería necesario definir diez nuevos canales para permitir al nuevo gestor de colas el acceso a todos los demás gestores.

Figura 6. **Cinco plataformas distintas cada una con un gestor de colas y cuatro definiciones de canales**



Esta fue la función primaria de los agentes de mensajes: aliviar la complejidad de la topología de la red y proporcionar enrutamiento y formateo simple de mensajes.

Figura 7. Cinco gestores de colas configurados utilizando un agente de mensajes (como *MQSeries Integrator*)



Las características más relevantes de *MQSeries Integrator* versión 2.0 son.

- **Descripciones de mensajes.** El soporte de descripción de mensajes significa que los desarrolladores no necesitan definir el formato de los mensajes. La composición de los mensajes se define utilizando estructuras de C y Cobol, clases de Java mensajes de XML, etc.
- **Flujos de mensajes.** El flujo de mensajes representa la secuencia de operaciones realizadas sobre un mensaje con base en las reglas correspondientes al formato y al contenido del mensaje. Los flujos de mensajes están compuestos por nodos de procesos y conectores. Los nodos de procesos actúan como unidades de trabajo

en un flujo de mensajes. Son simplemente procedimientos que realizan alguna acción específica. Los conectores enlazan los nodos de procesos dentro de un flujo, creando una secuencia lógica de acciones.

- **Enrutamiento.** *MQSeries Integrator* proporciona una colección de verificadores de sintaxis. Los filtros que utilizan estos verificadores chequean el formato del mensaje. Al evaluar el contenido del mensaje el agente de mensajes es capaz de direccionar al mensaje por la ruta adecuada. Los filtros para los mensajes pueden ser definidos por el usuario.
- **Publicación y suscripción.** Para poder aprovechar las características de publicación y suscripción de *MQSeries Integrator* es necesario comprender dos elementos básicos: publicadores de mensajes y subscriptores de mensajes.

Las aplicaciones cliente pueden suscribirse a un tema de mensajes (mensajes de interés común para productores y consumidores) o a un contenido de los mensajes. El agente de mensajes hace coincidir el tema de un mensaje publicado con una lista de clientes suscritos a ese tema. El publicador y el subscriptor no necesitan conocerse mutuamente. Los publicadores sólo saben los temas a los que pertenecen sus mensajes, y los subscriptores sólo conocen los temas a los que se registraron por interés propio. Los temas pueden clasificarse en árboles. En donde cada nivel del árbol se hace más específico que el nivel anterior.

**Tareas de usuario.** El centro de control de *MQSeries Integrator* permite al diseñador definir los mensajes y flujos de mensajes que especifican el tipo de lógica de negocio que se desea implementar. Consecuentemente, los conjuntos de mensajes y flujos de mensajes deben asignarse a un agente específico en una topología dada. Una vez que las definiciones y asignaciones han sido realizadas el sistema está listo para el desarrollo. El centro de control proporciona una interfase gráfica, de arrastrar y soltar (*drag-and-drop*) para realizar tareas de diseño, configuración, administración y operabilidad.

Sumario de los pasos comunes en la implementación de un ambiente con *MQSeries Integrator*:

- Planear la infraestructura requerida para el ambiente del agente de mensajes.
- Instalar el agente de mensajes: *MQSeries Integrator* siguiendo la infraestructura planificada.
- Configurar el dominio del agente de mensajes y la topología.
- Definir los conjuntos de mensajes y los flujos de mensajes.
- Asignar los conjuntos de mensajes y los flujos de mensajes al agente correspondiente.
- Definir los temas de publicación y suscripción y las políticas de control.
- Verificar que el dominio del agente de mensajes sea operacional.
- Controlar y administrar las operaciones del dominio operacional.
- Las aplicaciones existentes y las nuevas proveen el envío de mensajes y la lógica de enrutamiento y transformación de mensajes.

### 4.3. IBM MQSeries Workflow

En un sistema basado en **flujos de trabajo** (*Workflow*) se conoce el estado de las actividades del negocio instantáneamente gracias a un modelo basado en sucesos de activación. Estos sucesos o eventos se utilizan automáticamente para otras activaciones y la información que proporcionan es trasladada al departamento adecuado o es almacenada para su uso posterior. Con *MQSeries Workflow* se puede crear un modelo de procesos que automaticen el sistema en el manejo del personal, datos, aplicaciones y procesos del negocio u organización, incluyendo asociados externos que se conectan por medio de la Internet, intranets y extranets.

*MQSeries Workflow* es un sistema administrador de flujos de trabajo que permite definir, administrar y ejecutar los procesos del negocio a través de un software que construye una representación de la lógica de los flujos de trabajo. Un sistema de administración de flujos de trabajo consta de tres partes principales:

- a. **Una función de definición de procesos** con la que se modela los flujos de trabajo.
- b. **Una función de control**, que maneja el flujo de trabajo cuando está en funcionamiento.
- c. **Interfaces** que permiten a los usuarios y aplicaciones realizar las tareas específicas del negocio dentro del modelo del flujo de trabajo.

#### **4.3.1. Función de definición de procesos**

Esta parte de *MQSeries* Workflow se utiliza durante la construcción del flujo de trabajo y permite.

- Diseñar los procesos del negocio gráficamente.
- Definir el esquema de la empresa: personal, organizaciones, papeles de cada uno y niveles.
- Registrar los programas o aplicaciones.
- Definir las estructuras de datos.
- Definir la topología de funcionamiento.

Cuando el modelo del flujo de trabajo (procesos del negocio) está listo para funcionar, es tarea de la función de control de *MQSeries* Workflow administrar su funcionamiento.

#### **4.3.2. Función de control**

- Administrar el funcionamiento del flujo de trabajo
- Navegar a través de todo el modelo
- Notificar al administrador de retrasos
- Asignar las tareas individuales al personal adecuado si es necesario involucrar al equipo de trabajo
- Invocar a las aplicaciones relacionadas automáticamente
- Almacenar cada acción en la bitácora de auditoría para propósitos de evaluación

### 4.3.3. Interfaces

*MQSeries Workflow* soporta cierto número de aplicaciones e interfaces que conectan a los usuarios con el sistema de flujos e integra aplicaciones existentes y nuevas que pueden estar corriendo en una amplia gama de plataformas. *MQSeries Workflow* es un sistema cliente / servidor y tiene componentes en el cliente y en el servidor que son responsables de las distintas tareas del flujo de trabajo.

Las siguientes tareas y tipos de procesos pueden definirse con la ayuda de *MQSeries Workflow*:

- Definir y documentar los procesos del negocio.
- Controlar las actividades del personal dentro de los procesos.
- Automatizar aquellas actividades que no requieren control humano.
- Monitorear la operatibilidad de los procesos.
- Refinar y actualizar los procesos conforme a los cambios en las necesidades del negocio.
- Dirigir los procesos ya definidos hacia nuevos caminos.
- Integrar y alinear los procesos del negocio.
- Integrar las aplicaciones.
- Separar el flujo de procesos de la lógica de las aplicaciones.
- Respalda los procesos de control de calidad, auditoria, productividad, etc.
- Rápida y segura asimilación de los cambios.

*MQSeries Workflow* hace más fácil para el equipo de trabajo el manejo efectivo de los procesos proporcionando la información que se necesita, cuando se necesita. Los empleados pueden trabajar con aplicaciones amigables del tipo *front-end*, que se conectan con el núcleo de aplicaciones del negocio.

#### **4.3.4. Construyendo un modelo de flujo de trabajo**

El modelado de los procesos de negocio involucra la definición de procesos, incluyendo las actividades de la red de trabajo. Además, deben tomarse en cuenta a los participantes en los flujos de trabajo (equipos de trabajo y aplicaciones) y la tecnología necesaria para la implementación.

Una forma adecuada de iniciar el modelado consiste en resolver preguntas tales como:

- ¿Qué actividades pertenecen a un proceso?
- ¿Cuál es la secuencia en la que dichas actividades deben llevarse a cabo?
- ¿Es posible realizar algunas de estas actividades en paralelo?
- ¿Qué unidades de la organización están involucradas?
- ¿Qué aplicaciones están involucradas en el chequeo de la información?
- ¿Qué tipos de datos están involucrados?
- ¿Cuáles son las reglas del negocio que controlan el proceso?

Las respuestas a estas preguntas proveen la base para el modelo del flujo de trabajo. Con *MQSeries Workflow* es posible crear un esquema gráfico y definir esta información.

#### **4.3.5. Crear el modelo**

Con *MQSeries Workflow* se dibuja un modelo de procesos con sus distintos tipos de actividades. Además, se definen las propiedades para todos los componentes que forman parte del flujo de trabajo, como la organización, el personal, la información, los programas y las fuentes de tecnología necesarias.

Si la definición de un proceso se torna muy compleja se pueden utilizar subprocesos. La ventaja de esto es que se pueden utilizar los subprocesos en otros procesos. Es posible empezar por definir los subprocesos y con ellos construir los procesos involucrados en el modelo. Además, los procesos *MQSeries Workflow* proporcionan una serie de objetos que ayudan a una definición consistente del flujo de trabajo.

- Suceso de programa. Se usa para definir un programa extraído de una lista de programas, que un usuario inicia durante la ejecución de *MQSeries Workflow*.
- Suceso de proceso. Define un subproceso que forma parte del proceso en donde ocurrió el evento (suceso).
- Bloque de sucesos. Define una colección de actividades que pueden repetirse hasta que se produzca una condición de salida. El bloque es utilizado para definir un ciclo de actividades. También se definen bloques de sucesos en modelos complejos.
- Actividades. Son tareas que pueden definirse para ser ejecutadas en secuencia o en paralelo.

#### **4.3.6. Añadir lógica a los procesos**

Si el orden en el que las actividades inician es importante para los procesos, se puede controlar añadiendo conectores de control (enlaces) entre las actividades. Cuando el proceso está corriendo las condiciones definidas en los conectores son usadas para determinar cuáles actividades deben iniciarse y cuáles no.

#### **4.3.7. Añadir datos al flujo de trabajo**

Los datos que viajan entre las actividades de un proceso se definen con los conectores de datos en el diagrama del proceso. Para los datos que deben estar disponibles para los procesos que están corriendo se deben definir las propiedades de las estructuras de datos.

#### **4.3.8. Asignar personal a los procesos**

Luego de definir los procesos y sus actividades, se deben asignar los miembros del equipo de trabajo que deben efectuar las actividades durante el funcionamiento del flujo de trabajo. Se puede definir la organización y los roles que existan en ella. Existen varias opciones para asignar las actividades al personal.

- Asignar la actividad a la persona que inicia el proceso.
- Asignar la actividad al grupo de personas que tienen un papel en particular o que trabajan en un departamento en particular, o ambos, es decir que tienen un rol específico en un departamento específico.
- Asignar la actividad usando la información de entrada proporcionada por la misma actividad.
- Asignar la actividad a la persona que inicia otra actividad dentro del flujo de trabajo.

#### **4.3.9. Añadir programas al flujo de trabajo**

En el diagrama de procesos se definen las propiedades de las aplicaciones, herramientas y programas que forman parte del flujo de trabajo. Una aplicación es iniciada cuando alguien inicia la actividad correspondiente o, también, puede iniciarse automáticamente si está definida de esta manera.

La aplicación puede pertenecer a otra estación de trabajo o a otro servidor del sistema que podría estar usando un sistema operativo distinto.

#### **4.3.10. Poner a funcionar el modelo**

Cualquiera que esté autorizado puede controlar los procesos que están ejecutándose. Es posible iniciar, interrumpir y reiniciar procesos. También se pueden cambiar las asignaciones de trabajo, trasladando la actividad o tarea a alguien más. De esta manera se acelera la ejecución de ciertos procesos si alguna persona tiene demasiadas tareas y alguien más tiene muy pocas o ninguna.

*MQSeries Workflow* envía notificaciones automáticamente a las personas especificadas si los procesos, actividades y notificaciones no son completadas en un tiempo definido. Esta es otra forma con la que se asegura la velocidad de la ejecución de los procesos.

Se puede monitorear el estado de un proceso mientras está corriendo. Observar el progreso en el trabajo puede disminuir el tiempo de respuesta hacia los empleados y los clientes.

*MQSeries Workflow* almacena bitácoras de procesos en ejecución en un archivo. En esta bitácora se almacenan sucesos importantes que ocurren mientras un proceso está corriendo, por ejemplo, las veces que las actividades se realizan de principio a fin. Esto permite evaluar la eficiencia de los procesos.

## 4.4 MSMQ (*Microsoft Message Queue Server*)

El servidor de colas de mensajes de Microsoft (MSMQ), conocido por su nombre en código: Falcon, es un sistema de mensajería diseñado con un único propósito en mente: comunicación electrónica rápida, segura y sincrónica. Garantiza la entrega de los mensajes, la notificación de recibido y el control de transacciones.

### 4.4.1. ¿Qué es un mensaje en MSMQ?

Un mensaje es una unidad de información o datos que es enviada desde un proceso corriendo en una computadora hacia otro proceso corriendo en la misma o en una computadora distinta dentro de la red de trabajo.

Los sistemas de mensajería (*message oriented middleware*) reducen la complejidad envuelta en el desarrollo de aplicaciones que abarcan múltiples sistemas operativos y protocolos de comunicación. Proveen interfaces que se extienden a través de diversas plataformas y redes y, de este modo, ocultan a los desarrolladores los detalles de cada sistema operativo e interfase de red involucrados. Esto incrementa la flexibilidad de una arquitectura de comunicación al permitir a las aplicaciones intercambiar mensajes con otros programas sin tener que saber en qué plataforma o procesador reside la otra aplicación.

### 4.4.2. ¿Cuándo y dónde es necesario MSMQ?

- **Comunicación de almacenamiento y envío:** MSMQ permite a las aplicaciones enviar requerimientos a otras aplicaciones, en la misma o en distinta red, que podrían no estar corriendo o no estar disponibles al mismo tiempo.

- **Comunicación defensiva:** en sistemas de redes locales las aplicaciones están protegidas por MSMQ de pérdidas de datos cuando se produce un error en la red. MSMQ soporta picos de demanda y trabaja muy bien cuando la red funciona correctamente.
- **Procesamiento concurrente:** MSMQ permite a las aplicaciones enviar requerimientos hacia varios destinos sin esperar por cada respuesta; los mensajes son procesados en paralelo por las aplicaciones que recibieron los mensajes y cuando todas las respuestas se han recibido son trasladadas a la aplicación de origen.
- **Bitácora de comunicación:** cuando se habilita la opción de journal de MSMQ, automáticamente se almacena una copia de cada mensaje manejado en una cola de journal. Esto facilita las tareas de auditoría, recuperación y monitoreo de mensajes.
- **Comunicación no orientada a la conexión (no se establece una ruta específica para el envío de mensajes):** con MSMQ las aplicaciones que reciben mensajes pueden redireccionarlos hacia otras aplicaciones simplemente enviando el mismo mensaje a una cola diferente. O enviando el mismo mensaje las veces que sea necesario hacia cada uno de los destinos. Es indiferente si las aplicaciones están o no corriendo en cada plataforma destino.

#### 4.4.3. Algunas características importantes de MSMQ

- **Entrega asegurada de mensajes:** las aplicaciones que mantienen un intercambio de mensajes entre ellas no tienen que estar conectadas siempre a la red al mismo tiempo. MSMQ garantiza que los mensajes serán entregados tan pronto como las conexiones de la red se restablezcan y la aplicación destino solicite los mensajes recibidos.

- **Comunicación asincrónica:** una vez que una aplicación cliente envía un mensaje hacia una aplicación destino, MSMQ permite que la aplicación cliente realice otras tareas sin tener que esperar por una respuesta de la aplicación destino.
- **Soporte transaccional:** MSMQ soporta transacciones y puede integrarse fácilmente con el servidor transaccional de Microsoft (*Microsoft Transaction Server, MTS*). Cuando MTS está presente, sus servicios automáticamente participan en cualquier transacción de MSMQ.
- **Una sola entrega, en orden:** MSMQ garantiza que cada mensaje será entregado una y sola una vez y que será recibido en el orden en el que fue enviado.
- **Servicios de rutas para mensajes:** estos servicios le proporcionan a la aplicación cliente la posibilidad de enviar mensajes hacia su destino utilizando las rutas de menor costo. El administrador únicamente debe definir el costo de cada ruta, y MSMQ automáticamente calculará la ruta más económica para el mensaje. Esta búsqueda de rutas es capaz de eliminar puntos específicos de fallas. MSMQ es capaz de esquivar problemas en las rutas de la red. Esta característica es de mucha ayuda cuando se trabaja con redes poco seguras como la Internet.
- **Servicio de notificación:** aunque MSMQ permite que el intercambio de mensajes sea asincrónico liberando a la aplicación cliente, en algunos casos, la aplicación cliente puede necesitar la notificación de que el mensaje enviado ha sido recibido satisfactoriamente. Para estos casos, MSMQ provee la opción de que la aplicación cliente reciba las confirmaciones de recibido, de que otra aplicación las maneje, de que sean ignoradas estas notificaciones o de que sean almacenadas en una cola de journal para propósitos de auditoría.

#### 4.4.4. Configuración dinámica

- Escalabilidad
- Soporte para *ActiveX*: además de la interfase proporcionada, Microsoft también provee para una interfase de *ActiveX* que permite al usuario acceder la mayor parte de las funciones de MSMQ.
- Integración con el sistema de seguridad de *Windows 2000*: esto permite que el control de acceso a las colas sea manejado por el subsistema de seguridad de *Windows 2000*.
- Soporte para MAPI, MS Exchange y otros productos: además de una conexión con MS Exchange y una interfase MAPI, Microsoft y **Level 8 Systems** han desarrollado un producto llamado **FalconMQ** que extiende la funcionalidad de MSMQ a plataformas como AS/400, Unix, MVS, OS/2, y UNISYS. *FalconMQ* también provee operabilidad con los productos *MQSeries*.

#### 4.4.5. Descripción del funcionamiento de MSMQ

MSMQ puede realizar la entrega de mensajes de dos formas: los mensajes expresos son almacenados en memoria, mientras que los mensajes recobrables son almacenados en el disco duro. En ambos casos, las áreas utilizadas para el almacenamiento se definen como colas. Es posible asignarle prioridades a los mensajes; el envío de cada mensaje se basa primero en la prioridad de la cola y segundo, en la prioridad del mensaje.

En un esquema de comunicación construido con MSMQ todos los sistemas interoperan dentro de una arquitectura MSMQ (*MSMQ enterprise*). La arquitectura está subdividida en sitios. Los sistemas dentro de un sitio no tienen que usar el mismo protocolo de comunicación, ni siquiera es necesario que se comuniquen directamente con cada uno. Cada sitio está conectado con los otros sitios dentro de la arquitectura por medio de enlaces de sitio. Los administradores pueden crear y asignar estos enlaces. MSMQ escoge la ruta disponible que proporcione el menor costo para enlazarse con otro sitio.

Todos los sistemas dentro de una arquitectura MSMQ obtienen la información de un lugar común de almacenamiento, llamado *MSMQ Information Store* (MQUIS). Este lugar común de almacenamiento es una base de datos distribuida que guarda la topología de la arquitectura, sus atributos, la información de la computadora y la información de las colas. Las aplicaciones basadas en MSMQ pueden consultar el lugar común de almacenamiento y obtener las propiedades allí almacenadas.

Existen dos tipos de colas en MSMQ. Las colas públicas son aquellas que están publicadas en el lugar común de almacenamiento (MQUIS) y son replicadas a través de toda la arquitectura MSMQ. Por lo tanto, pueden ser localizadas por cualquier computadora dentro de la arquitectura. Las colas privadas por otro lado no son publicadas en el lugar común de almacenamiento y pueden ser accesadas únicamente por los sistemas que tengan acceso al nombre completo o formato de la cola.

Todos los sistemas que están presentes en una arquitectura MSMQ pueden clasificarse en: servidores, clientes dependientes y clientes independientes. Los servidores pueden subdividirse en: controlador primario del sitio (PSC), controlador de respaldo del sitio (BSC) y servidor MSMQ de enrutamiento.

El controlador primario del sitio contiene información acerca de la configuración de la arquitectura y las claves de autenticación de mensajes. El controlador primario del sitio, que debe instalarse uno por cada sitio, contiene información acerca de las computadoras y colas en el sitio así como también las funciones básicas de un servidor MSMQ de enrutamiento. Los servidores MSMQ de enrutamiento soportan enrutamiento dinámico y envío de mensajes de almacenamiento y envío. Permiten a las computadoras utilizar distintos protocolos para comunicarse. El controlador de respaldo del sitio proporciona recuperación en caso de que el controlador primario falle. Almacena una réplica de solo lectura de la base de datos del controlador primario.

Los clientes independientes se instalan en máquinas con *Windows 95/98* o *Windows 2000*. Pueden crear y modificar colas localmente y enviar y recibir mensajes. También pueden crear colas y almacenar mensajes en la computadora local sin sincronizarse con un Servidor de MSMQ.

Las funciones de los clientes dependientes son muy parecidas a las de los clientes independientes excepto que los clientes dependientes no pueden funcionar sin el soporte sincrónico de los servidores.

- Toda la arquitectura MSMQ puede manejarse por medio del explorador MSMQ. Este provee una interfase para administrar todas las máquinas en el ambiente MSMQ desde un punto centralizado de control.

## 4.5. Consideraciones de la planeación

La comunicación por medio de mensajes es una forma de comunicación de programa a programa. Las aplicaciones se comunican entre ellas enviando y recibiendo mensajes utilizando colas, sin necesidad de tener un enlace privado, lógico y dedicado.

Los siguientes puntos deben analizarse detenidamente cuando se está considerando una solución de integración por medio de mensajería entre sistemas. Es decir, que todavía no se sabe con exactitud si el problema de comunicación que se enfrenta puede resolverse con una arquitectura que permita el intercambio de mensajes entre aplicaciones.

- Las aplicaciones residen en múltiples plataformas de distinto tipo.
- La comunicación asincrónica es muy importante.
- Se desea construir un esquema de comunicación que permita el enlace efectivo con asociados de negocios constantemente.
- Es necesario integrar los sistemas internos para aumentar el valor y el uso de la información que se maneja.
- Es necesario intercambiar información entre sistemas que manejan lenguajes (formatos de mensajes) distintos.
- Se desea conectar en línea dos o más sistemas para realizar transacciones.

Los siguientes puntos deben estudiarse cuando ya se determinó que la integración por medio de mensajería es la solución correcta para el problema de comunicación que se enfrentan:

- **¿Qué herramienta de comunicación utilizar?** Los capítulos 3 y 4 describen algunas herramientas que pueden utilizarse para el intercambio de mensajes. También se presenta una forma rudimentaria de establecer la comunicación, en donde se desarrollan programas o módulos para el intercambio de mensaje y no se utiliza una herramienta especializada. La decisión sobre la herramienta que se utilizará debe hacerse tomando en cuenta, al menos, los siguientes puntos.
  - El presupuesto con el que se cuenta para implementar la solución.
  - La entrega asegurada de mensajes.
  - Las plataformas en donde deberá correr la herramienta seleccionada.
  
- **¿Qué *Message Broker* utilizar?** Es necesario que cumpla con, al menos, las siguientes características.
  - Proceso de mensajes de forma aislada, es decir, independiente de las aplicaciones del usuario.
  - Construcción de mensajes, por medio del enriquecimiento y generación de datos.
  - Aplicación de distintos formatos y reformateo para permitir la comunicación entre sistemas que manejen distinta mensajería.
  - Enrutamiento inteligente, basado en contenido o en reglas definidas.
  - Seguridad.
  
- Los recursos humanos con los que se cuenta.
- Los recursos tecnológicos
- El tiempo que se desea emplear para la implementación de la solución.
- ¿Qué formatos de mensajes se utilizarán en el esquema completo?

- ¿En qué puntos del esquema los mensajes deben ser persistentes? Es decir, deben ser almacenados para asegurar su recuperación en caso de fallos, aunque esto signifique aumentar el tiempo de viaje de los mensajes.
- ¿En qué puntos del esquema es más importante la velocidad de transmisión de los mensajes que la confiabilidad de que no se pierdan?

Estos son solo algunos puntos que deben considerarse, sin embargo, la selección de la mejor opción depende de cada proyecto que se analice.



## **5. ARQUITECTURA EN UN CASO REAL Y OTROS EJEMPLOS<sup>1</sup>**

### **5.1. Descripción del caso**

La compañía de teléfonos TransTel tiene sucursales en la capital y en el interior del país; cada una de estas sucursales tiene principalmente dos funciones: como punto de venta y como centro de captación. La primera función de la sucursal se dirige a atraer e ingresar nuevos clientes y la segunda función es recolectar los pagos de teléfono que los clientes llegan a realizar a la sucursal.

El objetivo de este proyecto para TransTel es eliminar de la mayoría de sus sucursales la función de centro de captación, para que los recursos, humanos, tecnológicos, etc., que están dedicados a esta tarea puedan reducirse (lo que implica una reducción de costos que es siempre atractiva) o enfocarse a la otra función principal de la sucursal que es el punto de venta. La meta es entonces conseguir más usuarios del servicio de telefonía dirigiendo los recursos de la sucursal hacia las ventas.

Para conseguir esta meta es necesario trasladar la función de centro de captación a una empresa que se dedique a eso, en este caso la mejor opción es un banco. El banco cuenta con los recursos y la experiencia necesarios para realizar esta tarea. El costo por transacción para TransTel se reduce debido a que recibir dinero es la tarea principal del banco, por lo tanto, tiene optimizados todos los procesos que esta tarea involucra o, al menos, los realiza de una forma más eficiente y menos costosa que la forma en que se realizan actualmente por TransTel.

---

<sup>1</sup> Los nombres utilizados en los casos descritos en este capítulo son ficticios.

Por otro lado, el Banco del Pacífico, como una estrategia de mercado para atraer nuevos clientes y mantener los que ya tiene, constantemente moderniza los servicios que ofrece y abre nuevos canales para que sus clientes puedan acceder a estos servicios de una forma más cómoda y rápida. El canal que el Banco del Pacífico desea explotar actualmente es el de la banca móvil. Al banco le interesa que sus usuarios puedan acceder sus cuentas desde un teléfono celular, y realizar algunas transacciones comunes, como el pago de tarjeta de crédito, consulta de saldos, transferencias entre cuentas (solo del Banco del Pacífico) y estado de cuenta.

El proyecto de TransTel de utilizar al Banco del Pacífico como recolector de los pagos de teléfono también es atractivo para el banco debido a que ofrece un nuevo servicio a sus clientes y, además, obtiene a TransTel como cliente nuevo lo que significa una o más cuentas de ingresos muy significativas para el banco.

De la misma manera, el proyecto del Banco del Pacífico de utilizar los teléfonos celulares de TransTel para que los clientes realicen transacciones bancarias también es atractivo para TransTel debido a que cada vez que el usuario utilice el servicio de banca móvil tendrá que hacer una llamada y, por lo tanto, estará consumiendo el producto que TransTel ofrece.

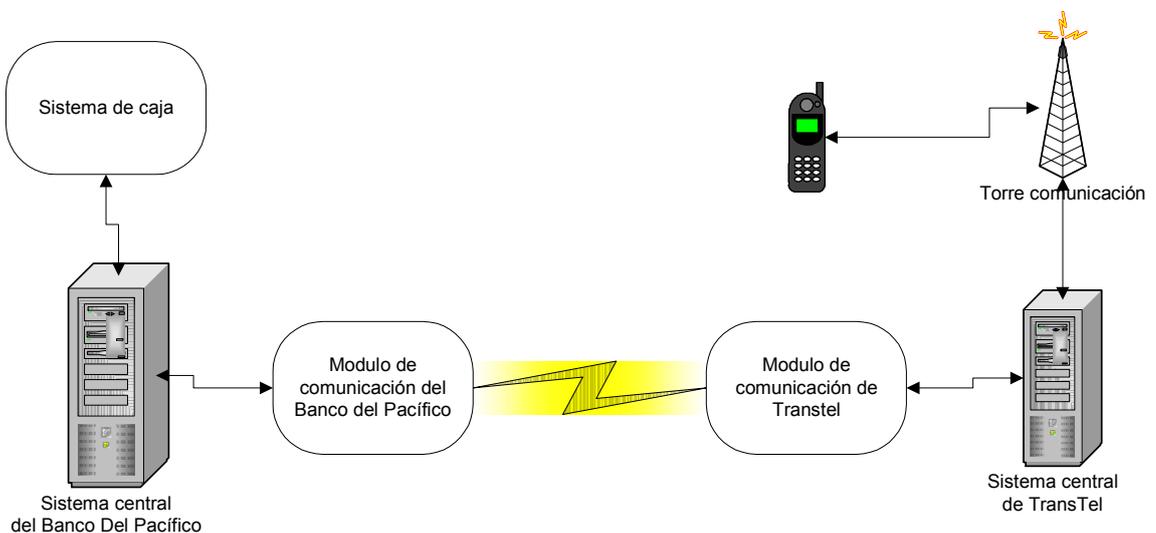
## **5.2 Sistemas involucrados y relaciones entre aplicaciones**

Es necesario anotar que en el proyecto completo de comunicación entre el Banco del Pacífico y TransTel se tienen dos tipos de transacciones:

- Las transacciones que se generan en el sistema de caja del banco: consultas de saldo de teléfono y pagos de teléfono.
- Las transacciones que se generan en cualquier teléfono celular de TransTel.

Los requerimientos y respuestas que se generen para resolver estas transacciones deben pasar por los sistemas que se muestran en la figura 8

Figura 8. **Esquema muy simplificado de los sistemas involucrados**



El proyecto de integración debe dirigirse a resolver la comunicación entre el sistema central del banco y el sistema central de compañía de teléfonos. Se deben resolver básicamente los siguientes puntos.

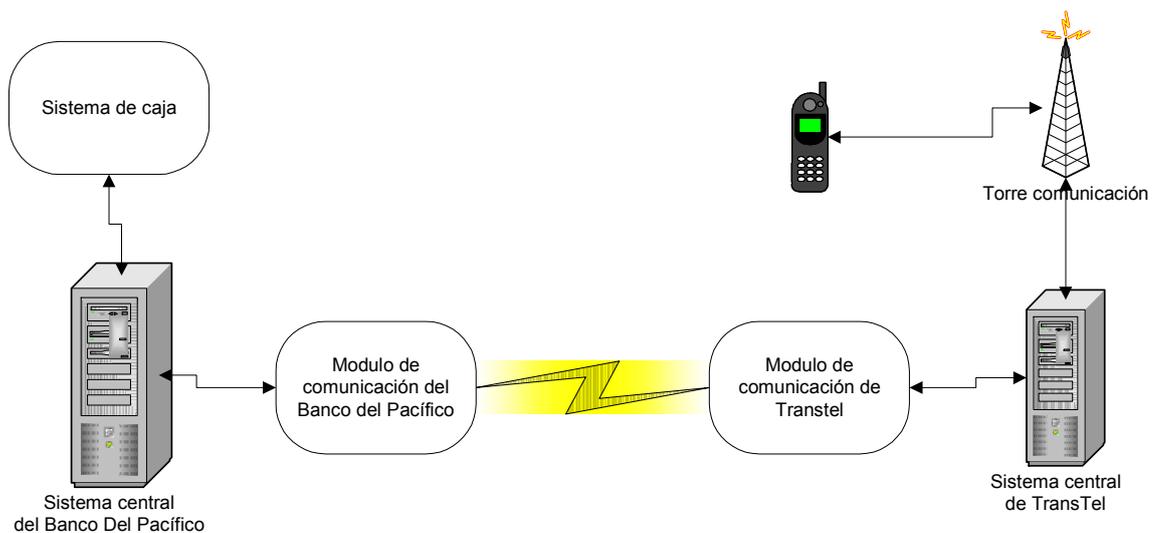
- Proporcionar el módulo de comunicación para el banco y el que corresponde a TransTel. Estos dos módulos pueden observarse en la figura 8.
- Debido a que se trata con transacciones se debe asegurar que éstas se ejecuten como unidades de trabajo, lo que implica que los módulos de comunicación deben garantizar la entrega de los mensajes y evitar que se dupliquen.

- Al conectarse desde un teléfono celular para realizar transacciones bancarias, la conexión se hace a un *web server* en la central del banco. Es decir que el proceso corresponde al de abrir un sitio de internet, los mensajes que se generan en este proceso están en formato XML.
- El sistema central del Banco del Pacífico corre en un *mainframe* en donde las aplicaciones utilizadas están desarrolladas principalmente en Cobol; para una compatibilidad más fácil entre las aplicaciones y los mensajes, el banco utiliza mensajes formados con campos de longitud fija.
- Como los mensajes en ambos extremos son distintos, se necesita un nodo que se encargue de traducir los mensajes de un formato a otro para que las aplicaciones en cada extremo no reciban mensajes que no comprenden y para que no se deba programar la transformación de mensajes en cada extremo y para cada mensaje.
- Los módulos deben permitir la comunicación en línea y el traslado de mensajes en forma *batch*.

### 5.3. Arquitectura de comunicación

Cada nodo en la figura 9 que representa a *MQSeries* se muestra independiente del sistema central correspondiente, sin embargo, físicamente existen dentro del servidor que hospeda al sistema central. Cada nodo *MQSeries* está compuesto por varios componentes: un manejador o controlador de colas de mensajes, un conjunto de colas de mensajes y canales de comunicación.

Figura 9. **MQSeries independiente del sistema central**



El proceso representado por el círculo A es un enlace entre *MQSeries* y el sistema central del Banco del Pacífico. El proceso B de manera semejante es el conector o enlace entre el sistema central de TransTel y *MQSeries*. Los dos procesos tienen una estructura parecida que se explica a continuación.

Cada proceso esta compuesto por dos aplicaciones

1. La primera, bastante sencilla, sólo recibe requerimientos o respuestas desde el sistema central y los traslada hacia *MQSeries* para que sean enviados al otro extremo.
2. La segunda, recibe mensajes de *MQSeries* y los traslada al sistema central para que éste los procese, su forma algorítmica es la siguiente:

- *Recibir mensaje de una cola de MQSeries.*
- *Procesar la transacción (consulta, pago, etc.)*
- *El programa queda a la espera de una respuesta del sistema central.*
- *Convertir la respuesta recibida en un mensaje de MQSeries.*
- *Poner el mensaje de respuesta en una cola de MQSeries para que sea enviado al integrador.*

Gracias a que se tiene *MQSeries Integrator* en el medio, las aplicaciones A y B pueden olvidarse de la transformación de mensajes ya que recibirán los mensajes en la forma en que el sistema central correspondiente entiende y, también, pueden enviar los mensajes de respuesta y los requerimientos en su propio formato sin preocuparse de que el otro extremo entienda o no los mensajes enviados.

## **5.4 Configuración de *MQSeries Integrator***

Por el nodo del *MQSeries Integrator* pasan todos los mensajes. Existen dos tareas esenciales que el integrador debe realizar: enrutar los mensajes y reformatearlos. Para mostrar los procesos que el integrador lleva a cabo, se utiliza como ejemplo una transacción de consulta de saldo de una cuenta monetaria. Esta transacción se genera en el teléfono celular del usuario que desea saber el saldo de su cuenta monetaria del Banco del Pacífico.

- El usuario se conecta al *HTTP Server* desde su celular y envía la consulta de saldo.
- El *HTTP Server* envía la consulta al sistema central de TransTel, el mensaje enviado tiene formato XML.
- El sistema central de TransTel recibe el requerimiento y, al identificar que la transacción no le pertenece, la envía al proceso B (ver figura 9).
- El proceso B recibe el requerimiento y lo envía a *MQSeries*.
- *MQSeries* de TransTel, recibe el mensaje en la cola salida y lo traslada al *MQSeries Integrator*.
- La consulta del saldo de la cuenta está ahora, en una cola de *MQSeries* dentro del integrador en formato XML.
  - El integrador saca el mensaje de la cola.
  - Identifica el mensaje como una consulta de saldos en formato XML.
  - Identifica el origen: TransTel y el destino: Banco del Pacífico.
  - Le aplica al mensaje un formato XML para separar y reconocer los campos.
  - Genera un mensaje con los campos obtenidos del mensaje XML. El mensaje generado utiliza un formato de campos de longitud fija.
  - El integrador pone el mensaje generado en la cola correspondiente al destino identificado: Banco del Pacífico.
- El mensaje, ya transformado, llega al *MQSeries* del Banco del Pacífico.
- El proceso A, que está a la espera de mensajes, recibe la transacción de consulta de saldo en la cola de entrada. Saca el mensaje de la cola y lo envía al sistema central del banco para que sea procesado.
- El proceso A queda a la espera de la respuesta.

- La respuesta llega desde el sistema central del banco, se construye el mensaje de *MQSeries* y se coloca en la cola de salida del manejador de colas del banco.
- El mensaje es enviado al integrador.
- El Integrador recibe el mensaje en formato de campos de longitud fija.
  - El Integrador saca el mensaje de la cola.
  - Reconoce el mensaje como la respuesta de una consulta de saldos.
  - Identifica el origen: Banco del Pacífico y el destino: TransTel.
  - Le aplica al mensaje un formato de campos de longitud fija, previamente definido.
  - Transforma el mensaje a uno con formato XML: toma los campos del mensaje recibido y le aplica los formatos y encabezados que corresponden a un mensaje XML.
  - El integrador coloca el mensaje reformateado en la cola que corresponde al destino identificado: TransTel.
- El mensaje de respuesta a la consulta de saldos, en formato XML, llega finalmente al *MQSeries* de TransTel.
- El proceso B toma la respuesta, saca el mensaje de la cola, y lo envía al sistema central de TransTel.
- El sistema central de TransTel traslada la respuesta al *HTTP Server* que se encarga de mostrar el saldo solicitado al usuario del teléfono celular que inició la transacción.

## 5.5. Otros ejemplos reales

La arquitectura de los ejemplos siguientes puede ampliarse para integrar más sistemas a cada uno de los esquemas, ya que las herramientas y el diseño permiten que la arquitectura crezca y, por lo tanto, la información que un sistema maneje se puede poner a disponibilidad de otros sistemas que la necesiten.

Al observar una integración entre dos empresas puede parecer que resultaría más fácil resolver la comunicación entre ambas con aplicaciones específicas (a la medida) para cada una. Sin embargo, una de las ventajas de utilizar productos que resuelven la integración, como *MQSeries*, es que la arquitectura permite añadir otros sistemas con un esfuerzo menor al que sería necesario si se añadieran más sistemas a una solución con aplicaciones a la medida.

### 5.5.1. Una compañía distribuidora de artículos para el hogar

Esta solución para la integración de sistemas se aplica de manera eficiente para una cadena de tiendas que vende artículos para el hogar: muebles, aparatos eléctricos, etc. La cadena de tiendas tiene varias sucursales en Guatemala pero es originaria de México. La cadena resuelve varios problemas utilizando *MQSeries*. Los más importantes son:

- **Inventario:** la comunicación con sucursales le permite a la cadena observar la demanda y la existencia de los productos en cada tienda. Esto es útil para tener disponibilidad de productos para los clientes y para no tener un inventario muy grande, sino más bien un inventario de acuerdo a las necesidades de la región en la que se ubica cada tienda.

- **Consolidar contabilidad:** el enlace con sus sucursales le permite realizar cierres consolidados para mantener una contabilidad más clara y precisa.

### **5.5.2. Un banco y una institución estatal**

El ejemplo en este caso está compuesto por un banco y por una institución estatal que hace captación de dinero. El Departamento de Tránsito debe captar las multas y los pagos que, por una u otra razón, deben pagar algunos conductores y/o dueños de vehículos; sin embargo, el Departamento de Tránsito no tiene la infraestructura ideal para realizar esta tarea. Conforme la captación se hace mayor el proceso se vuelve más complicado para dicha institución debido a que debe emplear más recursos (humanos, de equipo y espacio) para llevarla a cabo. Para que esta actividad no traiga incomodidad a la institución y a los usuarios, el Departamento de Tránsito decide dejar la tarea de captación al banco.

El banco habilita entonces las transacciones necesarias para recibir los pagos de que corresponden al Departamento de Tránsito. Cada vez que el banco recibe uno de estos pagos, envía, por medio de la arquitectura de *MQSeries*, una confirmación al Departamento de Tránsito del pago realizado. El Departamento de Tránsito, al recibir la confirmación, acredita el pago a la persona correspondiente.

Para el usuario de este servicio es necesario que exista la entrega asegurada, para que el Departamento de Tránsito no tome las acciones que comúnmente toma cuando no realizan los pagos a tiempo, por ejemplo, suspender las licencias, recoger los vehículos, etc.

### 5.5.3. Tres sistemas internos de un banco

El siguiente ejemplo involucra tres sistemas importantes de un banco:

- Sistema central o *core* bancario
- Sistema de cajeros automáticos
- *Call center*

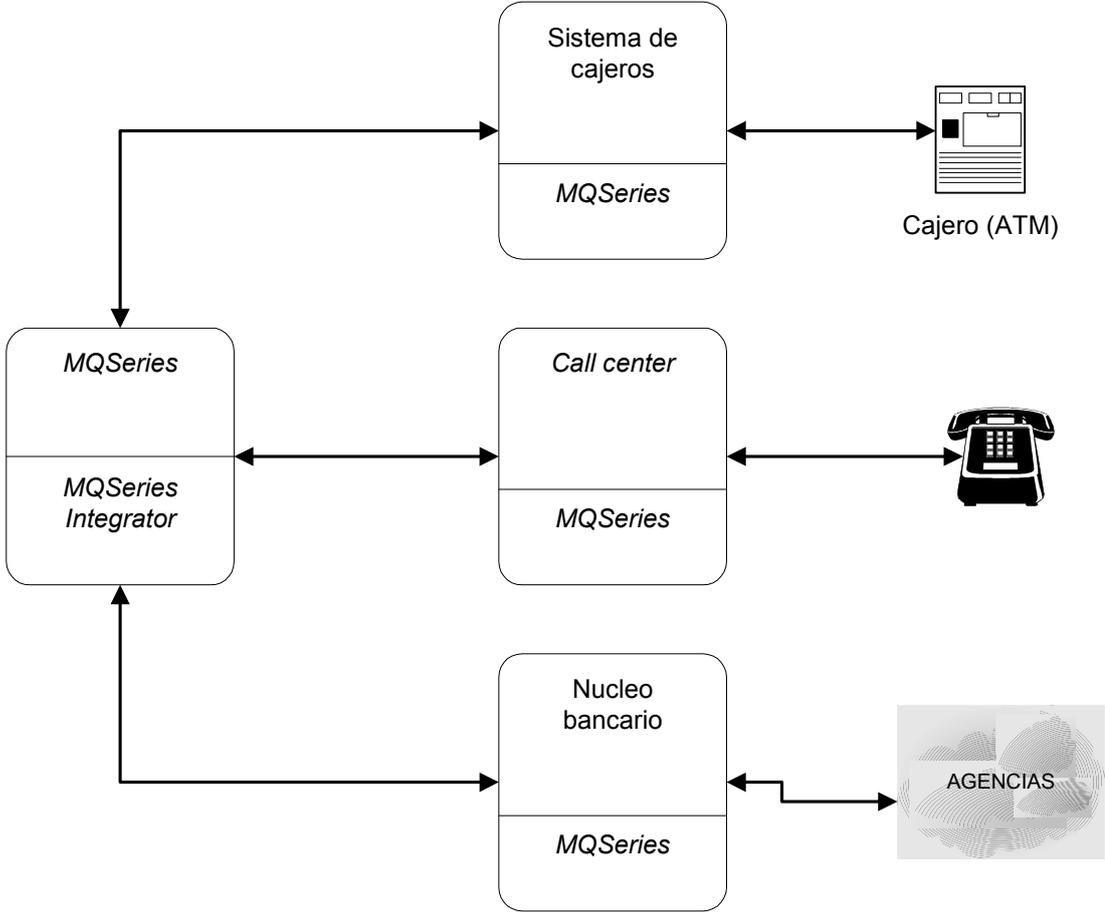
Los requerimientos del sistema de cajeros y del *call center* se envían al integrador. Éste se encarga de transformar los mensajes y enviarlos al *core* bancario, éste resuelve y envía la respuesta (si es necesario) al integrador.

La tarea principal de *MQSeries* en este esquema (ver figura 10) es la de proporcionar un canal seguro y eficaz de comunicación entre los sistemas. Las tareas del integrador son.

- a. Identificar el origen y el destino de los mensajes
- b. Aplicar el formato de origen del mensaje
- c. Construir un mensaje de salida que corresponda al destino del mensaje
- d. Enviar el mensaje transformado al destino correspondiente

Si el núcleo bancario estuviera dividido en varios sub-sistemas, al utilizar la integración de este tipo el núcleo sería percibido como uno solo por los otros sistemas (cajeros, *call center*, etc.).

Figura 10. MQSeries proporcionando un canal seguro



## CONCLUSIONES

1. La integración de negocios es efectiva cuando los sistemas involucrados pueden colaborar entre ellos, compartiendo la información que manejan, para alcanzar sus objetivos.
2. Las empresas se ven empujadas a cambiar (reestructurando sus sistemas internos o asociándose con otras empresas) por sus propias necesidades de hacer frente a la competencia, de mejorar su servicio al cliente y de agilizar sus procesos de manufactura y distribución. Este cambio significa muchas veces integrar sus sistemas de información internos o enlazarlos con los sistemas de sus asociados de negocios.
3. Existen varias formas de hacer frente a una integración de negocios, la selección de la mejor solución para un caso determinado, depende de las condiciones en las que se debe trabajar y de los recursos con los que se cuenta.
4. Cuando la tecnología se utiliza al máximo y es superada, para que no se convierta en una limitante, deben efectuarse cambios en ella que permitan alcanzar los objetivos y realizar las tareas propuestas.

## RECOMENDACIONES

1. La integración de negocios debe contemplar la adaptación y conectividad que existe entre los sistemas empresariales para poder implantar nuevos esquemas de negocio con suficiente agilidad y flexibilidad.
2. Una integración debe planificarse con suficiente anticipación tomando en cuenta:
  - a) Las plataformas involucradas
  - b) Las aplicaciones o sistemas de información que residen en cada plataforma
  - c) Los formatos de mensajería que utilizan esos sistemas de información
  - d) Las herramientas más adecuadas para resolver la integración
  - e) Los recursos con los que se cuenta
3. Se debe llegar a un acuerdo sobre los formatos de los mensajes que van a viajar a través de los sistemas así como sobre las reglas que controlarán las rutas que los mensajes deben seguir dentro de la arquitectura de comunicación.
4. Aunque en este documento se analiza a fondo únicamente la familia de productos *MQSeries* de IBM, se recomienda considerar para cualquier proyecto de integración otros productos presentes en el mercado como MSMQ y Biztalk de Microsoft, Tuxedo, NEONet, etc., que también pueden resolver efectivamente una integración de negocios.

5. La arquitectura de comunicación implementada para una integración, debe ser flexible para adaptarse a los requerimientos futuros que se vayan generando para los sistemas involucrados. Estos requerimientos se producen debido a que en cada sistema la disponibilidad de la información aumenta lo que significa que se abren nuevas opciones para el uso de esta información.

## BIBLIOGRAFIA

1. **BET. *Wokflow, Groupware, gestión de procesos...***  
[http://www.bet.es/texto\\_wf\\_gestionprocesos.htm](http://www.bet.es/texto_wf_gestionprocesos.htm). 30/08/2000.
2. **BET. *Wokflow, Groupware, gestión de procesos...***  
[http://www.bet.es/texto\\_wf\\_gestionprocesos.htm](http://www.bet.es/texto_wf_gestionprocesos.htm). 30/08/2000.
3. **BET. Las tres "C". Clientes, competencia y cambio.**  
[http://www.bet.es/texto\\_las\\_tres\\_c\\_empresa.htm](http://www.bet.es/texto_las_tres_c_empresa.htm). 30/08/2000.
4. **CÁRDENAS, Jorge E. Información y simplificación de procesos.**  
<http://www.extra-net.net/forum/missatges/10.html>. 10/09/2000.
5. **IBM. *Business integration*.** <http://www.software.ibm.com/big>. 9/08/2000.
6. IBM Latinoamérica. “Integración de la Empresa”. **Soluciones de *software* para la integración de la empresa.**(Venezuela),(1):1-9. 1999.
7. IBM Latinoamérica. “Las soluciones IBM para la integración de la empresa aumentan el éxito de las fusiones y adquisiciones corporativas”. **Soluciones de *software* para la integración de la empresa.**(Venezuela),(1):1-9. 1999.
8. IBM Latinoamérica. “Transformando las relaciones con clientes a través de las soluciones para la integración de la empresa”. **Soluciones de *software* para la integración de la empresa.**(Venezuela),(1):1-13. 1999.
9. IBM Latinoamérica. “Construyendo la autopista financiera”. **Soluciones de *software* para la integración de la empresa.**(Venezuela),(1):1-9. 1999.

10. IBM Latinoamérica. "Aumente su ventaja competitiva en "e-business" con las soluciones IBM para la integración de la empresa". **Soluciones de software para la integración de la empresa.**(Venezuela),(1):1-8. 1999.
11. IBM United Kingdom Limited. "**Executive overview**". *Software solution for business integration.* (England), (1): 1-8. 2000.
12. IBM United Kingdom Limited. "**Getting the edge in e-business with IBM solutions for Web integration**". *Software solution for business integration.* (England), (1): 1-8. 2000.
13. IBM United Kingdom Limited. "**Transforming customer relationships with business integration**". *Software solution for business integration.* (England), (1): 1-12. 2000.
14. IBM *MQSeries Manuals published in 2000.* <http://www-4.ibm.com/software/ts/MQSeries/library/manuals99/>. 16/01/2001.
15. IBM *MQSeries: Library.* <http://www-4.ibm.com/software/ts/MQSeries/library/mqpdfs.html>. 16/01/2001.
16. IBM *Software: Transaction Systems: MQSeries Workflow: Overview.* <http://www-4.ibm.com/software/ts/MQSeries/workflow/>. 01/02/2001.
17. IBM *United Kingdom Limited.* "*MQSeries Workflow*", *MQSeries family of products*, (England). 2001.
18. IBM *Corporation.* *System Administration. MQSeries Manuals.* Segunda edición. IBM *United Kingdom Laboratories*, 2001.
19. IBM *Corporation.* *Student Notebook. MQSeries System Administration.* Quinta edición. Junio, 2001.

20. KOONTZ, Harold y Heinz Weihrich. **Administración. Una perspectiva global.** 10a ed. McGraw-Hill, 1995.
21. LANG, Jay H. "IBM MQSeries Integrator V2.0 *The next generation message broker*". *Independent Reports.* IBM Corporation. Enero, 2000.
22. **LEVEL 8. What Is middleware?**  
[http://www.level8.com/index.cfm?handle=pbg\\_whatismiddleware.](http://www.level8.com/index.cfm?handle=pbg_whatismiddleware)  
05/02/2001.
23. **MESSAGE** *Queuing in Windows 2000.* [www.messageq.com](http://www.messageq.com). 13/01/2001.
24. **MQSeries** Version 5.1 *Administration and Programming Examples.*  
<http://www.redbooks.ibm.com/abstracts/sg245849.html>. 15/01/2001.
25. **SUMATORIA, S.R.L. Administración y optimización de la cadena de distribución.** <http://www.sumatoria.com.ar/>. 5/10/2000.
26. **SURESH** Raj, Gopalan. Microsoft *Message Queue Server (MSMQ).*  
<http://www.execpc.com/~gopalan/mts/msmq.html>. 13/01/2001.
27. **USING** MSMQ. [msdn.microsoft.com](http://msdn.microsoft.com). 13/01/2001.

## ANEXOS

### Modelo de comunicaciones SNA

La configuración de redes de IBM hoy en día consiste en dos arquitecturas separadas que provienen de un origen en común. Antes de la existencia de las redes de computadoras contemporáneas, el sistema de arquitectura de red de IBM, (SNA, por sus siglas en inglés de “*System Network Architecture*”), imponía las reglas del desarrollo de redes de computadoras, pero a menudo esto es visto como lo tradicional, o como el legado de SNA. Con el crecimiento del uso de computadoras personales, estaciones de trabajo (*workstation*), y las arquitecturas cliente / servidor, hizo que IBM dirigiera su estrategia a cubrir la necesidad de tener arquitecturas de red basadas en nodos de trabajo, de ese redireccionamiento se creó el *Advanced Peer-to-Peer Networking* (APPN), y el *Advanced Program-to-Program Computing* (APPC).

#### SNA Tradicional

SNA fue desarrollado en el transcurso de los años 70, éste permite a un supercomputador ejecutar algo conocido como *Advanced Communication Facility/Virtual Telecommunication Access Method* (ACF/VTAM), el cual sirve como concentrador para una red SNA. ACF/VTAM es responsable de establecer todas las sesiones de trabajo, y de activar y desactivar todos los recursos. En este tipo de sistema de trabajo, los recursos son predefinidos antes de ser usados, eliminando así el tráfico que se provoca al hacer solicitudes a través de mensaje tipo *broadcast*.

## Arquitectura SNA

El modelo SNA se divide en capas, las cuales son:

Control de enlace de datos:

DLC, por sus siglas en inglés *Data-link control*, define muchos protocolos de comunicación, incluyendo el control de enlace de datos sincrónico (SDLC), que se utiliza para la comunicación de tipo jerárquica; y el protocolo conocido como *Token Ring Network communication protocol*, que se utiliza para la comunicación entre nodos miembros de una red de área local, LAN.

Control de ruta:

*Path control*, incluye funciones como el ruteo de la información y el servicio de segmentación y reensamblado de datagramas (SAR).

Control de transmisión:

*Transmission control* se asegura de que la información saliente del emisor de datos, llegue al receptor con total seguridad, haciéndolo a través de servicios de encriptación y desencriptación de datos.

Capa de control de flujo de datos:

*Data flow control* se encarga de administrar los procesos de solicitud y respuesta de información, determina de quién es el turno de comunicarse, maneja grupos de mensajes, e interrumpe flujos de datos cuando se le solicita.

## Servicios de presentación

Esta capa especifica los algoritmos de transformación de los datos, que serán los que traduzcan los datos de un formato a otro, también coordina la forma en que se comparten los recursos y sincroniza las operaciones transaccionales.

## Capa de servicios de transacción

Provee de servicios de aplicación, en forma de programas que implementan el procesamiento distribuido y/o el manejo de servicios.

SNA no define un protocolo específico para el manejo de la capa física, ya que se asume que éste será implementado tomando en cuenta otros estándares.

## **APPN**

La segunda generación del SNA de IBM es el *Advanced Peer-to-Peer Networking*, en el que se hizo el cambio de tener un SNA basado en comunicaciones jerárquicas, con un supercomputador centralizando todas las operaciones, por un sistema basado en nodos de trabajo. En el corazón del APPN, se encuentra una arquitectura que soporta comunicaciones punto a punto, servicios de directorio y ruteo entre dos o más sistemas APPC, los cuales no están conectados directamente a la misma red de trabajo.

## **Componentes APPN**

En adición al ambiente APPN, el modelo de red punto a punto de SNA especifica tres conceptos clave: las unidades lógicas (LUs), comunicación de programa a programa avanzada (APPC), y nodos tipo 2.1. Cada uno de éstos juega un papel importante en la forma de establecer la comunicación entre los diferentes nodos SNA, dentro del contexto de las interredes basadas en el modelo punto a punto del SNA.

Las unidades lógicas (LU) 6.2 controlan la comunicación punto a punto dentro de un ambiente SNA, además, LU 6.2 soporta comunicación entre programas en un ambiente de procesamiento distribuido, así como entre nodos de diferentes tipos. El APPC permite a las aplicaciones SNA comunicarse directamente entre sí, no importando si se encuentran en diferentes ubicaciones, también, APPC provee un conjunto de protocolos y convenciones de programación que implementa el LU 6.2.

Los nodos tipo 2.1s (T2.1) son entidades lógicas que permiten la comunicación directa del ambiente SNA, con nodos periféricos que son capaces de soportar T2.1. Las entidades T2.1 facilita la comunicación punto a punto, proveyendo el soporte de transporte de datos, para los niveles de comunicación soportada por el APPC. Además, un T2.1 contiene un punto de control para nodos periféricos (PNCP), que combina las funciones tradicionales de una unidad física (PU) del SNA tradicional, y un punto de control (CP), que es por donde se establece la conexión entre los nodos en el ambiente SNA de la segunda generación.

## **El modelo de referencia TCP/IP**

El modelo de referencia TCP/IP, surgió junto con la red de computadoras más antigua que se conoce, la ARPANET. La ARPANET era una red de investigación patrocinada por el Departamento de Defensa de los Estados Unidos, la cual al final conectó a cientos de universidades e instituciones del gobierno usando líneas telefónicas rentadas; la ARPANET se volvió tan popular y eficiente que hoy en día se conoce como la Internet, y por esto TCP/IP, llamado así por las iniciales de sus dos protocolos primarios, es tan popular.

## **Arquitectura TCP/IP**

La arquitectura TCP/IP también está compuesta por varias capas, las cuales son:

### La capa de interred

Es el eje que mantiene unida toda la arquitectura TCP/IP, y su misión es permitir que los nodos introduzcan paquetes en cualquier red y los hagan viajar de forma independiente a su destino. La capa de interred define un formato de paquete y protocolo oficial, llamado IP (por sus siglas en inglés *Internet Protocol*), entonces, la tarea de la capa de interred es entregar paquetes IP al lugar a donde deben ir.

### Capa de transporte

Está sobre la capa de interred, y se encarga de que al momento de hacer una transmisión de datos exista una conversación entre las partes involucradas de los nodos origen y destino.

Dentro de esta capa se definen dos protocolos de extremo a extremos, TCP y UDP; siendo el primero de ellos el protocolo de control de transmisión, y el cual es un protocolo confiable orientado a la conexión, que se encarga de que una corriente de *bytes* originadas en una máquina se entregue sin errores en cualquier otra máquina de la red. El segundo protocolo es conocido como protocolo de datagrama de usuario, es un protocolo sin conexión, no confiable y se utiliza para aplicaciones que no necesitan asignación de secuencia ni control de flujo alguno.

#### La capa de aplicación

Esta capa está por encima de la capa de transporte y contiene todos los protocolos que se consideran como de alto nivel, tal es el caso de los protocolos de terminal virtual (*TELNET*), de transferencia de archivos (*FTP*) y de correo electrónico (*SMTP*), que son de los más antiguos que hay.

#### Capa de nodo de red

Bajo la capa de interred existe un gran vacío, un vacío del que el modelo TCP/IP no hace mucha referencia, ni de lo que allí sucede, y simplemente se limita a indicar que un nodo se ha de conectar a la red haciendo uso de algún tipo de protocolo, de tal modo que pueda enviar por ella paquetes de IP.

El servicio TCP/IP se obtiene haciendo que tanto el emisor de datos como el receptor creen puntos terminales llamados conectores (*sockets*). Cada conector tiene una dirección conocida como número de conector, este número está formado por la dirección IP de la terminal, y un número de 16 bits local a esa terminal, que se conoce como puerto. Entonces, para obtener el servicio TCP/IP es necesario establecer una conexión entre un conector de la máquina emisora y un conector de la máquina receptora.

