



PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE
ESCUELA DE INGENIERIA

**BRIDGING THE GAP BETWEEN THE
SCIENCE OF RETAIL REVENUE
MANAGEMENT AND ITS APPLICATION**

DIEGO JOSÉ FUENTES HENRÍQUEZ

Tesis presentada a la Dirección de Investigación y Postgrado
como parte de los requisitos para optar al grado de
Magister en Ciencias de la Ingeniería

Profesor Supervisor:

JUAN CARLOS FERRER ORTIZ

Santiago de Chile, Agosto 2008

© MMVIII, DIEGO JOSÉ FUENTES HENRÍQUEZ



PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE
ESCUELA DE INGENIERIA

**BRIDGING THE GAP BETWEEN THE
SCIENCE OF RETAIL REVENUE
MANAGEMENT AND ITS APPLICATION**

DIEGO JOSÉ FUENTES HENRÍQUEZ

Miembros del Comité:

JUAN CARLOS FERRER ORTIZ

SERGIO DANIEL MATURANA VALDERRAMA

HUGO RODRIGO MORA CARRASCO

IGNACIO ANTONIO CASAS RAPOSO

Tesis presentada a la Dirección de Investigación y Postgrado
como parte de los requisitos para optar al grado de
Magister en Ciencias de la Ingeniería

Santiago de Chile, Agosto 2008

© MMVIII, DIEGO JOSÉ FUENTES HENRÍQUEZ

A mi familia.

AGRADECIMIENTOS

Quisiera agradecer a todas las personas que me apoyaron en el desarrollo de esta tesis. En primer lugar, a mi familia, por su paciencia y respaldo. También al profesor Juan Carlos Ferrer, por su guía, oportunidades de aprender y ayuda que me brindó.

No puedo dejar de mencionar a Hugo Mora, quien en los momentos más difíciles me dio el ánimo y la guía necesaria para salir adelante.

También debo agradecer a los miembros de Pricing UC: Diego Oyarzún, Pedro León Eduardo Flores, Francisco Olivares y Felipe Aros. Cada uno de ellos hizo su aporte a la presente investigación, y en conjunto forman un tremendo grupo humano y académico.

INDICE GENERAL

AGRADECIMIENTOS	iv
INDICE DE FIGURAS	vii
INDICE DE TABLAS	viii
RESUMEN	ix
ABSTRACT	x
1. Introducción	1
1.1. Revisión de la Literatura	3
1.2. Hipótesis	6
1.3. Estructura de la Tesis	6
2. Modelos de Revenue Management	7
2.1. Estimación de la demanda	7
2.2. Optimización	8
3. Diseño del Sistema	11
3.1. Base de Datos	11
3.1.1. Dimensiones.	11
3.1.2. Series.	12
3.1.3. Eventos, promociones y factores exógenos.	14
3.2. Pricing Core	16
3.3. Interfaz de Usuario	18
3.4. El proceso de toma de decisiones	18
3.4.1. Objetos comunes de interfaz	20
3.4.2. Módulos de la Aplicación	22
3.4.3. Flujo	25

4. Conclusiones	26
ANEXO A. Glosario de términos	28
A.1. GMROI	28
A.2. Object Relational Mapping	29
A.3. Decision Support Systems	29
A.4. Tecnologías Web	31
A.5. Business Intelligence	33
BIBLIOGRAFIA	35

INDICE DE FIGURAS

3.1 Visión general del sistema.	12
3.2 Arquitectura del Pricing Core.	16
3.3 Proceso de análisis de productos.	19
3.4 Proceso de fijación de reglas del negocio.	20
3.5 Ventas de un producto bajo efectos de precio, inventario y promociones. . . .	21
3.6 Flujo.	25

INDICE DE TABLAS

1.1 Compañías que ofrecen software de pricing para retail.	5
3.1 Dimensiones y agrupaciones posibles.	11
3.2 Categorías de series, y su aditividad.	12

RESUMEN

Esta investigación busca establecer una conexión entre la teoría y la práctica de Retail Revenue Management, diseñando un Sistema de Apoyo a la Toma de Decisiones que se basa en la experiencia obtenida de proyectos de investigación aplicada en distintos retailers. De esta manera, se espera que los usuarios puedan tomar mejores decisiones de precio e inventario, usando las herramientas que proveen los avances de Revenue Management y de las Tecnologías de Información; esto permite un análisis eficiente de los datos que se encuentran almacenados en sus sistemas. Se detalla el esquema de datos, la arquitectura del sistema y la interfaz de usuario. El diseño ha sido exitosamente implementado en una cadena de retail chilena.

Palabras Claves: Sistemas de Apoyo a la Toma de Decisiones, Revenue Management, Pricing, Retail.

ABSTRACT

This research aims at establishing a connection between theory and practice of Retail Revenue Management by designing a Decision Support System based on the experience acquired from several applied research projects in various retailers. It is expected that, in this manner, users will be able to make better pricing and inventory decisions by using the tools provided by developments in Revenue Management and Information Technologies, whereby an efficient analysis of the data stored in their systems can be performed. The data schema, system architecture and user interface are described. This design has been successfully implemented in a Chilean retail chain.

Keywords: Decision Support Systems, Revenue Management, Pricing, Retail.

1. INTRODUCCIÓN

En los últimos años, se ha visto un aumento en el uso de modelos de Revenue Management (RM) en diversas industrias. Comenzando con las aerolíneas en los años 70, el desarrollo se ha extendido a hoteles, arriendo de automóviles, telecomunicaciones, retail, manufactura, comercio electrónico, etc. Según Talluri y Van Ryzin (2004), el RM posee dos áreas principales: (1) basado en cantidad (*Quantity-Based*), que gestiona la demanda por medio de racionar la disponibilidad de producto; y (2) basado en precio (*Price-Based* o *Pricing*), que mediante éste busca balancear la oferta con la demanda.

En el caso del retail, la literatura aborda principalmente el RM basado en precio (Bitran y Mondschein, 1997; Gallego y Van Ryzin, 1994; Smith y Achabal, 1998). El problema clásico es el que enfrenta un retailer que debe vender una cantidad fija de productos perecibles en un horizonte de ventas finito. Dado que es posible manejar la demanda a través de la política de precios (Bitran y Caldentey, 2003), el objetivo es determinar aquélla que maximice el ingreso esperado.

El Retail Revenue Management (RRM), ciencia que se dedica al estudio de RM aplicado a la industria del retail, ha extendido el problema clásico desarrollando modelos enfocados en aspectos tales como: competencia, comportamiento del consumidor, efectos de sustitución de productos, etc. Dichas extensiones han generado modelos cada vez más sofisticados y complejos de resolver, dificultando la aplicación directa de RRM en la industria.

El avance de las Tecnologías de Información (TI) ha posibilitado abordar la creciente complejidad de los modelos. De hecho, Talluri y Van Ryzin (2004) afirman que la tecnología es uno de los pilares fundamentales en el desarrollo de RM. Sin embargo, los autores han observado una brecha entre el RRM y su aplicación en la industria, ya que muchos ejecutivos del retail se basan principalmente en su intuición para tomar decisiones. Las razones de esta brecha se pueden clasificar como: (1) técnicas, (2) de ambiente interno y (3) de ambiente externo.

Dentro de las razones técnicas se encuentran las relacionadas con los datos y con los sistemas de información de las empresas. En el primer caso se observa que existen retailers que no cuentan con suficientes datos históricos, que no están agregados adecuadamente, los ejecutivos no tienen acceso a ellos, o simplemente no los almacenan. En el segundo caso, los sistemas de información generan reportes que no proveen información suficiente para la toma de decisión de los ejecutivos.

Las razones de ambiente interno consideran los factores encontrados al interior de la empresa. Por una parte, la gran diversidad de funciones que deben cumplir los ejecutivos les impide aplicar modelos de RRM. Por otro lado no existen incentivos para que ellos apliquen estos modelos.

Por último, las razones de ambiente externo se refieren a factores exógenos, es decir, aquellos que la empresa no puede controlar, agregando complejidad al proceso de toma de decisiones. La competencia es uno de los más importantes, debido a su alto impacto en la demanda.

El objetivo de este artículo es contribuir a la reducción de la brecha antes mencionada, incorporando en las empresas los avances de RRM a través del diseño de un Sistema de Apoyo a la Toma de Decisiones o DSS (*Decision Support System*, véase Anexo A.3). De esta manera, los ejecutivos lograrán: (1) consolidar datos dispersos de la empresa, que se encuentran en sus bases de datos, archivos de texto y conocimiento personal, (2) incorporar los avances en modelos de RRM, y (3) utilizar una interfaz amigable para ellos. De este modo, el sistema aborda las causas que generan la brecha entre la ciencia del RRM y su aplicación.

Como resultado de la investigación se definen tres pasos esenciales en el diseño de un DSS para RRM. Primero es necesario entender cómo integrar distintos modelos de RRM al sistema, lo que incluye determinar los datos necesarios y cómo obtenerlos. El segundo paso es detallar la arquitectura del sistema, que debe ser flexible para agregar nuevos modelos. Finalmente, se debe desarrollar una interfaz que permita a los usuarios

incorporar modelos sofisticados de RRM a su toma de decisiones. Si bien se mencionan tecnologías específicas, el diseño presentado es independiente de la plataforma usada en su implementación.

1.1. Revisión de la Literatura

En el diseño del sistema existen tres áreas de investigación relevantes: (1) avances en modelos de RRM, (2) avances en DSS, e (3) implementación de RRM en empresas.

Talluri y Van Ryzin (2004) realizan una completa revisión de los distintos modelos de RM, basados en cantidad y precio, además consideran elementos como pronósticos de demanda e implementación en diversas industrias, incluida el retail. Elmaghraby y Keskinocak (2003) y Bitran y Caldentey (2003) revisan modelos aplicados específicamente al retail (RRM).

Uno de los primeros trabajos en RRM es el realizado por Kincaid y Darling (1963), quienes plantean un modelo de tiempo continuo cuya demanda sigue un proceso de Poisson, y los potenciales compradores poseen un precio de reserva (máxima disposición a pagar) que sigue una distribución conocida por el retailer. Un modelo similar es usada por Bitran y Mondschein (1997), quienes dividen el horizonte de tiempo en períodos discretos y comparan el modelo planteado con el modelo de tiempo continuo, encontrando que las diferencias en los ingresos son bajas (menos de un 2.2%) si se escogen adecuadamente los momentos para realizar los cambios de precios. Un modelo determinístico es desarrollado por Smith y Achabal (1998), quienes cuantifican, entre otros efectos, la merma en las ventas cuando el inventario en tienda baja de un cierto nivel crítico.

Casi en paralelo con el desarrollo de Revenue Management (comienzo de los años 70), se han desarrollado los DSS. Power (2002) analiza los distintos tipos de DSS y discute metodologías de implementación. Arnott y Pervan (2005) y Arnott y Pervan (2008) realizan una revisión general de la literatura disponible a partir del año 1990. Los autores encuentran que, a pesar del gran aumento a nivel comercial de los DSS, el número de

investigaciones cayó en más de un 30% durante el período 2000-2004 comparado con el período 1995-1999. Sugieren como estrategia para mejorar la investigación en el área realizar estudios relevantes en la práctica y que éstos tengan una buena base teórica. Por otra parte, Bhargava et al. (2007) revisan el desarrollo de DSS basados en tecnologías Web. A pesar de los avances en el área, pocos artículos se enfocan en elementos de arquitectura de software y en proveer pautas para el diseño basado en evidencia empírica. Además, aún quedan por resolver desafíos tecnológicos, económicos y sociales.

Diversas investigaciones han reportado aumentos en los beneficios por la aplicación de modelos de RRM, debido a (1) los bajos márgenes de la industria, y (2) que cualquier mejora en éstos por medio del precio se traduce directamente en un aumento de la utilidad neta. Por ejemplo, la empresa ShopKo reportó un incremento de 24% en sus márgenes netos por el uso de un sistema de RRM para optimización de precios (Johnson, 2001), mientras que retailers como Gymboree, J.C. Penney, L.L Bean, Liz Caliborne, Safeway, y Walgreen incrementaron sus márgenes brutos entre un 5 y un 15% (Friend y Walker, 2001). El trabajo de Smith y Achabal (1998) muestra la implementación del modelo que desarrollan en tres empresas (dos exitosas), obteniendo aumentos en el margen bruto de hasta \$15 millones de dolares, siendo la disponibilidad y exactitud de la información determinantes en el éxito o fracaso. Heching, Gallego, y Van Ryzin (2002) analizan la información histórica de la temporada de primavera de 1993 de un retailer de ropa de mujer, encontrando un aumento potencial de ingreso de entre 2.4 y 13.6%.

Aprovechando este potencial, diversos proveedores de software ofrecen soluciones que permiten a los retailers aplicar RRM, especialmente pricing, en su toma de decisiones. En la Tabla 1.1 se listan algunos ejemplos. Tanto Elmaghraby y Keskinocak (2003) como Montgomery (2005) enumeran algunas características de estas soluciones, entre ellas: (1) identificación de oportunidades, (2) determinación de curvas de precios, (3) determinación de fechas de cambios de precios, y (4) capacidad de manejar categorías de productos sustitutos. A continuación se describen tres ejemplos de estos sistemas.

TABLA 1.1. Compañías que ofrecen software de pricing para retail.

Empresa	Web Site
ProfitLogic (Oracle)	http://www.oracle.com/profitlogic
DemandTec	http://www.demandtec.com
SAP for retail	http://www.sap.com/usa/industries/retail/large/optimization
SAS Revenue Optimization Suite	http://www.sas.com/industry/retail/rev_opt
JDA Markdown Optimization	http://www.jda.com/solutions/markdowns.html
PriceStrat (KSS Retail)	http://www.kssretail.com
Predictix	http://www.predictix.com

El primero es PROS, discutido por Valkov (2005), implementado en cincuenta empresas de once industrias diferentes. El autor destaca tres funciones claves en un software de pricing: (1) analítica, que explora datos históricos para descubrir oportunidades de incrementar las ganancias; (2) optimización de políticas de precios, complemento de la parte analítica; y (3) ejecución, que implementa los cambios de precios en los sistemas de información de la empresa.

El segundo software de pricing es MARK (comercialmente conocido como B_Line), descrito por Mantrala y Rao (2001), el cual obtiene políticas óptimas de precios aplicadas al retail, especialmente para productos de moda. El sistema posee dos módulos, con y sin determinación del inventario inicial. El modelo de demanda lo establecen los usuarios especificando la venta mínima, la más probable y la máxima para cuatro precios permitidos. Finalmente, se demuestra la ventaja de usar modelos de optimización en la determinación de las políticas de precios, encontrando beneficios hasta un 29% mayores respecto a los descuentos arbitrarios, además de aislarlos de errores al establecer de inventario inicial.

Finalmente, Cassaigne y Singh (2001) especifican la arquitectura de *PriceStrat*, software aplicado a la industria petrolera, supermercados y telefonía. El sistema posee tres partes principales: (1) módulo de pronósticos, (2) optimizador, y (3) módulo de aprendizaje. La base de datos también se divide en tres partes: (1) datos de la empresa, (2) base de conocimiento, (3) y datos de modelos.

Los trabajos nombrados anteriormente realizan descripciones generales de DSS que aplican modelos de RRM. Además las soluciones de software existentes, debido a su carácter comercial, en general no revelan elementos de diseño del mismo, como base de datos, arquitectura e interfaz de usuario, salvo el trabajo de Cassaigne y Singh (2001). Es por esto que el presente artículo busca realizar una descripción más detallada de un DSS que aplique modelos de RRM.

1.2. Hipótesis

A pesar de los avances en Tecnologías de Información (TI), especialmente las tecnologías Web y DSS (Power, 2002; Arnott y Pervan, 2008; Bhagarva et. al, 2007), existe relativamente poco desarrollo respecto a la aplicación de modelos de RM al retail que permitan un manejo eficiente de los datos y modelos matemáticos para modelar la demanda y optimizar precios e inventario (Cassaigne y Singh, 2001; Mantrala y Rao, 2001; Montgomery, 2005).

La hipótesis de este trabajo es que es posible especificar un diseño efectivo de un DSS de RM que apoye la toma de decisiones de precio e inventario óptimos a los ejecutivos del retail, decisión que involucra manejar gran cantidad de información de muchos productos y los modelos de demanda y optimización.

1.3. Estructura de la Tesis

En el capítulo 2 se muestran los modelos de RRM aplicables al sistema que se describirá posteriormente en el capítulo 3, detallando los datos necesarios (3.1), la arquitectura del sistema (3.2), y la interfaz de usuario (3.3). Finalmente en el capítulo 4 se exponen las conclusiones y perspectivas de avances futuros.

2. MODELOS DE REVENUE MANAGEMENT

La implementación de modelos de RRM considera dos pasos, descritos a continuación: (1) estimación de la demanda y (2) optimización de precios e inventario.

2.1. Estimación de la demanda

La generación de pronósticos precisos es determinante en el éxito de todo sistema de RM. Para ello se requiere una etapa de investigación previa, que defina modelos de demanda aplicables a los diferentes productos de cada empresa, además de metodologías de ajuste y pronóstico.

Los modelos determinísticos son ampliamente usados en aplicaciones prácticas, debido a que normalmente sólo se cuenta con información de ventas. En este contexto, se asume que la demanda está afectada por un conjunto de factores exógenos f_i , que abordan aspectos específicos del problema. La función de demanda normalmente es de forma multiplicativa: $D_{det} = f_1 f_2 \dots f_n$ (Mantrala y Rao, 2001; Smith y Achabal, 1998), ya que en la mayoría de los casos presenta un mejor ajuste que la forma aditiva (Smith, McIntyre, y Achabal, 1994). Se ha observado que existen cinco efectos principales que afectan la demanda de un producto, las que se incorporan como variables o factores en el modelo: estacionalidad, promociones, eventos, precio y surtido de inventario.

Una vez modelada la demanda se utilizan técnicas como análisis de regresión o métodos bayesianos (Montgomery, 1997; Bertsimas y Perakis, 2006; Aviv y Pazgal, 2005) para determinar el valor de cada parámetro o factor. Además, maximizando índices como el AIC (Akaike, 1974) o R^2 ajustado es posible encontrar modelos tales que todos sus factores sean estadísticamente significativos. Por otra parte, es posible aplicar el denominado enfoque jerárquico, que consiste en calibrar ciertos parámetros a un nivel de agregación de producto mayor (variables estacionales, promociones y eventos), y otros a un nivel menor (efecto precio).

A pesar de los buenos resultados de usar la modelación y las metodologías presentadas, se ha encontrado que éstas tienen al menos cinco limitaciones: (1) no existe información del comportamiento de los consumidores para precios distintos a los implementados; (2) tampoco se considera el comportamiento estratégico de los consumidores (Talluri y Van Ryzin, 2004); (3) en escasez de inventario, la demanda no corresponde a las ventas (Mantrala y Rao, 2001); (4) al momento de introducir un producto nuevo, puede que no exista otro similar para determinar su potencial comportamiento; y (5) existe una serie de factores exógenos, no considerados en los modelos, tales como sequía, bajas temperaturas, cambios macroeconómicos, etc. Es por esto que el desarrollo de los modelos de demanda continúa. Se han investigado nuevos enfoques, centrados en el consumidor en vez del producto, y se han usado metodologías como modelos de simulación o análisis de atributos (Van Ryzin, 2005). Este hecho debe estar presente en el diseño del DSS, el cual debe ser modular, permitiendo incorporar en forma directa los nuevos modelos y metodologías que vayan apareciendo.

El factor subjetivo, aportado por los tomadores de decisiones en base a su experiencia y conocimiento, es un complemento importante para cualquier modelo de demanda. Éste se puede incorporar al sistema permitiendo al usuario modificar el modelo calibrado, variando sus parámetros. También es posible hacer análisis de sensibilidad, a partir de la curva de demanda calibrada. Por ejemplo, el usuario puede variar la estrategia de precios para determinar los cambios en la demanda.

2.2. Optimización

El objetivo del problema es maximizar el beneficio esperado en el horizonte de ventas, a través de dos variables de decisión: inventario inicial a ordenar y estrategia de precios óptima. Una variante se presenta cuando la orden ya ha sido realizada, en cuyo caso el inventario se considera costo hundido, y se busca la política de precios que maximiza el ingreso. El horizonte de venta se divide en K períodos, cuya duración depende de la industria, pudiendo ser diaria, semanal, etc. De este modo se define $V_k(C)$ como el

ingreso máximo esperado con C unidades de inventario, quedando k períodos de venta. Por otra parte, sea $g(C)$ el costo de adquirir C unidades de inventario al inicio de la temporada. En el caso más simple, esta función es lineal, pero puede tomar otras formas si existen economías de escala o se agregan otros costos. Debido a que el horizonte de venta dura sólo unos meses, no se consideran elementos como costos de almacenamiento o factor de descuento, siendo éste igual a 1 (Bitran y Mondschein, 1997). Con esto, el problema que determina el inventario inicial es el siguiente:

$$Z = \max_C \{V_K(C) - g(C)\} \quad (2.1)$$

Sea $D_k(p)$ la demanda esperada para el período k a un precio p . La función $V_k(C)$ es como sigue:

$$V_k(C) = \max_{p_k} \{p_k \min\{C, D_k(p_k)\} + V_{k-1}(C - \min\{C, D_k(p_k)\})\} \quad (2.2)$$

s.a.

$$p_k \in \Pi \quad \forall k = K, \dots, 1 \quad (2.3)$$

Este es un problema de programación dinámica determinística, que debe resolverse hacia atrás en el tiempo. El caso estocástico es discutido por Bitran y Mondschein (1997). Las condiciones de borde son las siguientes:

$$V_0(C) = 0 \quad \forall C$$

$$V_k(0) = 0 \quad \forall k$$

En la restricción (2.3) del problema de optimización, Π denota al conjunto de precios factibles, dado por una serie de restricciones conocidas como reglas del negocio. Éstas se

definen a nivel estratégico y buscan alinear las políticas de precios con los objetivos de la empresa, permitiendo, entre otras cosas:

- Limitar la cantidad de cambios de precio posibles o la fecha de éstos.
- Definir un conjunto discreto de precios, por ejemplo los terminados en “90” o “9”, ampliamente utilizados debido a su impacto positivo en las ventas, especialmente en productos nuevos o cuyo precio no es conocido por el usuario (Anderson y Simester, 2003).
- Impedir aumentos en los precios, vale decir $p_k \leq p_{k-1}$.
- Posicionar el producto dentro de la misma empresa (según tamaño, para diferenciar marcas, etc.) o respecto a la competencia (e.g. precio siempre un 5% bajo la competencia).

Por lo tanto el sistema debe permitir a los usuarios resolver modelos de optimización, basado en las curvas de demanda calibradas y sujeto a las reglas del negocio previamente fijadas, que variarán según la empresa.

3. DISEÑO DEL SISTEMA

El sistema presentado se divide en tres componentes principales, tal como se muestra en la Figura 3.1. El primero es la base de datos, que consolida la información de la empresa. El segundo componente, denominado *Pricing Core*, implementa en sus distintos módulos los modelos de RRM que se incorporarán al sistema. Finalmente, la interfaz de usuario establece la comunicación del sistema con los tomadores de decisiones. En las siguientes secciones se describen con mayor detalle estos componentes.

3.1. Base de Datos

Los datos necesarios para el sistema se dividen en: (1) dimensiones (productos, locales, tiempo) o unidad de análisis; (2) series, que pueden pertenecer a cuatro categorías: ventas, precios, inventario y series de segundo orden; y (3) información de eventos, promociones y factores exógenos.

3.1.1. Dimensiones.

En la Tabla 3.1 se muestran las dimensiones y algunas agrupaciones comúnmente utilizadas. Los productos y locales se consideran de forma similar: es posible especificar uno, un grupo jerárquico establecido por la empresa, o un grupo definido por el usuario.

TABLA 3.1. Dimensiones y agrupaciones posibles.

Dimensión	Agrupaciones
Productos	Departamento
	Categoría
	Estilo
	Definidas por el usuario
Locales	Zona de precios
	Zona geográfica
	Definidas por el usuario
Tiempo	Año
	Mes
	Día

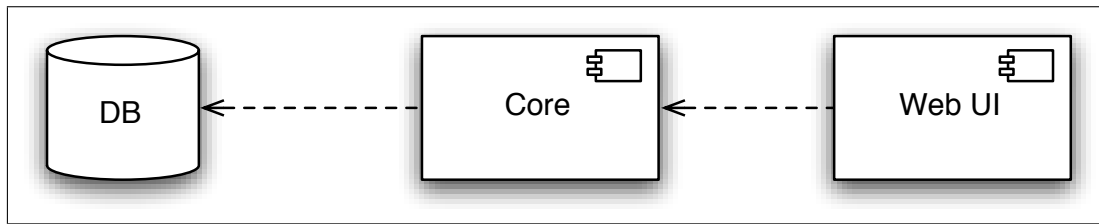


FIGURA 3.1. Visión general del sistema.

TABLA 3.2. Categorías de series, y su aditividad.

Categoría	Series	Aditividad		
		Producto	Local	Tiempo
Ventas	Física Valorada	sí	sí	sí
Inventario	Físico Valorado	sí	sí	no
Precios	de Sistema de Venta	no	no	no
2º orden	Margen % Contribución GMROI etc.	-	-	-

Un ejemplo es la zona de precios para la dimensión de locales, que permite al retailer fijar distintos precios para cada zona. Por su parte, la dimensión tiempo se define por medio de un rango más su agrupación o agregación temporal. También debe ser posible usar media móvil para suavizar la variabilidad de algunas series.

3.1.2. Series.

En la Tabla 3.2 se muestran las cuatro categorías de series, incluidas las de segundo orden, y su aditividad en las distintas dimensiones.

La primera categoría corresponde a las ventas, que constituyen la base para determinar la respuesta de los consumidores ante elementos estacionales, precios, promociones, etc (Montgomery, 2005). Esta información proviene normalmente del software de punto de venta (POS) que registra cada transacción efectuada. La categoría posee dos series:

(1) venta física, referida a la cantidad de producto vendida, y (2) venta valorada, que corresponde al ingreso. Ésta última se puede descomponer, según la disponibilidad de datos de la empresa, en valorada contado y a crédito (ingreso por pagos con el medio de financiamiento del retailer). Dicha distinción ha permitido detectar diversos efectos, por ejemplo una mayor sensibilidad al precio de los consumidores que pagan al contado. Las ventas poseen la propiedad de ser aditivas en todas las dimensiones.

La segunda categoría, el inventario, también posee dos series: (1) inventario físico, referido a la cantidad de productos almacenados, e (2) inventario valorado, referido al costo de estos. Disponer de esta información requiere de un gran espacio de almacenamiento, debido a la enorme cantidad de productos y locales que maneja un retailer (Elmaghraby y Keskinocak, 2003). Por esta razón resulta útil almacenar sólo los cambios de inventario. Aunque esto dificulta su obtención, tanto el volumen de los datos como el tiempo de respuesta se reducen considerablemente. Finalmente, cabe señalar que el inventario no es aditivo en la dimensión tiempo, puesto que no tiene sentido sumar el inventario en un rango de fechas, aunque sí es posible obtener un promedio.

La tercera categoría listada en la Tabla 3.2 corresponde a los precios, los cuales poseen forma de escalón al graficarlos con respecto al tiempo, debido a sus pocas variaciones. Se distinguen dos series de precios: (1) de sistema, encontrado en la base de datos de la empresa, y (2) de venta, calculado a partir del cociente entre la venta valorada y la venta física, utilizado cuando los primeros difieren de los efectivamente pagados por los clientes. Los precios no son aditivos en ninguna dimensión, aunque también es posible obtener un promedio.

Finalmente la Tabla 3.2 lista la categoría de series de segundo orden, calculadas a partir de las series anteriores. Ejemplo de ellas son el margen porcentual, la contribución, el GMROI (*Gross Margin Return On Investment*, véase Anexo A.1), etc. En la mayoría de los casos resulta útil precalcularlas, de modo de mantener un tiempo breve de respuesta al usuario. Su aditividad dependerá de cada una.

En la práctica se considera razonable contar con al menos dos años de historia. Aunque esto no permite suficiencia estadística, se puede usar un año para calibrar modelos de demanda y otro para validar el comportamiento del modelo. Por otro lado, cada serie debe estar al menor nivel de agregación posible. En el caso del tiempo, ésta debe ser diaria. A pesar de que existen aplicaciones con datos agregados semanalmente (Heching et al., 2002; Smith et al., 1994; Smith y Achabal, 1998), hay fuertes variaciones diarias que deben ser consideradas.

3.1.3. Eventos, promociones y factores exógenos.

Tanto los eventos como las promociones son acontecimientos que pueden afectar la demanda, pero se diferencian en que los primeros no pueden ser controlados por el retailer (e.g. navidad). Las promociones, en tanto, son manejadas por la empresa, y se clasifican en temáticas, de precio o respuesta a un determinado evento (e.g. promociones navideñas). Se realizan en un rango de fecha específico, y afectan a un conjunto de productos y locales, aunque en muchos casos aumentan o disminuyen la demanda de otros (efecto complementario o sustitución). Además poseen diversos focos, es decir, un subconjunto de los productos y locales en promoción son exhibidos a través de un medio (e.g. radio, televisión), durante el desarrollo de ésta. Normalmente el calendario de eventos y promociones no se encuentra sistematizado; en tal caso debe ser mantenido por los mismos usuarios.

Los factores exógenos son aquellos que afectan la demanda de un modo menos directo, tales como el clima, indicadores macroeconómicos (e.g. inflación, desempleo), etc. Su valor puede ser cualitativo (e.g. invierno muy frío) o numérico (e.g. % inflación mes de mayo).

Opcionalmente, es posible agregar más información al sistema, por ejemplo datos de la competencia o de los clientes. En el primer caso, se pueden conocer precios, promociones, participación de mercado, etc., a un nivel de detalle que varía según el tipo de retail. Su obtención se ha simplificado con el surgimiento del comercio electrónico,

dando paso a aplicaciones que extraen estos datos automáticamente (Baumgartner et al., 2005). Por otro lado, los retailers almacenan cada vez más información de sus clientes, lo que es clave para modelos de demanda orientados a los consumidores, donde influyen preferencias y hábitos de compra.

A causa de los diferentes esquemas y sistemas de base de datos en las distintas empresas, es conveniente tener una sola fuente de datos, creando lo que se denomina “una versión de la verdad” (*One Version of the Truth*). A partir de este esquema (relacional), se pueden aplicar técnicas como Inteligencia de Negocios (Anexo A.5), siguiendo la estrategia de Inmon (Jukié, 2006), permitiendo una rápida obtención de datos críticos, por ejemplo las series mencionadas a un alto nivel de agregación, o las de segundo orden.

La recolección de información implica la construcción de dos procesos ETL (*Extraction, Transformation, Loading*), lo que requiere participación activa de quienes se encargan de los sistemas de información de la empresa. El primer proceso recolecta los datos operacionales disponibles, para integrarlos formando una versión consolidada y almacenarlos en la base de datos del sistema. Se ha visto que el esfuerzo en este aspecto es mayor al que parece a simple vista, debido a las diversas fuentes de datos, transformaciones y verificaciones necesarias. Su funcionamiento debe ser al menos diario. El segundo proceso realiza los precálculos, y su frecuencia de funcionamiento dependerá del nivel de agregación temporal mínimo de esta información.

La precisión de los modelos de RRM depende de la calidad de los datos con que cuenta la empresa. Sin embargo, los autores han observado que muchas empresas no almacenan suficiente información histórica, o que ésta sólo es agregada. En otros casos los procesos organizacionales existentes generan datos imprecisos, por ejemplo niveles de inventario negativos o precios de sistema que difieren de los reales. Esto hace necesario un esfuerzo por parte de la empresa para (1) mejorar la precisión de sus datos, y (2) comenzar a almacenar nueva información, no disponible hasta ahora.

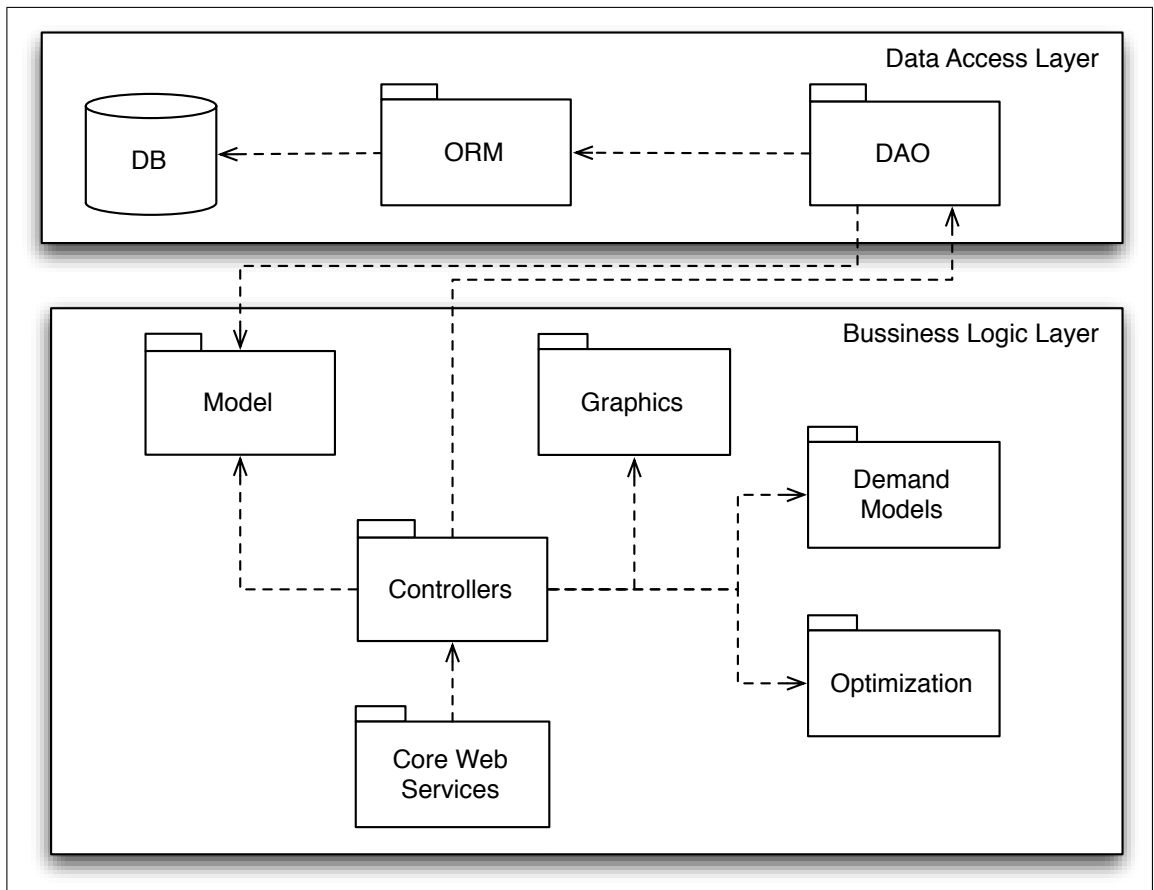


FIGURA 3.2. Arquitectura del Pricing Core.

3.2. Pricing Core

El denominado *Pricing Core* se encarga de la parte funcional del sistema. Su diseño, tal como se aprecia en la Figura 3.2, distingue dos capas principales: acceso a los datos, y lógica del negocio.

Previo al desarrollo de ambas capas, es necesario establecer el modelo orientado a objetos que incorpora las entidades y relaciones que requiere el sistema, formando la base para el transporte y almacenamiento de los datos. Su diseño considera los distintos elementos discutidos en la sección 3.1, como las dimensiones y series.

La capa de acceso a los datos abstrae el esquema de almacenamiento, lo que permite mantener o modificar diferentes bases de datos sin afectar la lógica del negocio. Para este propósito es recomendable usar el patrón de diseño DAO (*Data Access Object*, Sun Microsystems Inc. 2002) y una herramienta ORM (*Object Relational Mapping*, véase Anexo A.2) para persistir de modo transparente los objetos de la aplicación.

La capa de lógica de negocio contiene seis módulos principales: (1) modelo de objetos (ya descrito), (2) gráficos, (3) modelos de demanda, (4) optimización, (5) *controllers*, y (6) vista o interfaz.

Los gráficos facilitan el análisis de los datos, y son una buena forma de mostrar la intuición detrás de los resultados de los modelos de RRM. Aunque existen diversas librerías gráficas para distintas plataformas y tecnologías, es conveniente crear un proveedor genérico, con un modelo de objetos propio que genere independencia del resto de la aplicación, y por ende menor acoplamiento.

Los módulos de modelos de demanda y optimización implementan RRM en el sistema. El primero debe permitir calibrar y hacer pronósticos, mientras que el segundo debe incorporar no sólo la optimización, sino también la capacidad de fijar reglas de negocio. Dado que en este caso también existen variadas librerías que facilitan la calibración y optimización, nuevamente se recomienda generar independencia mediante un proveedor genérico.

Los *controllers* traducen los requerimientos que recibe de la vista en acciones para los distintos módulos ya mencionados. Cada solicitud de datos, gráficos, modelos de demanda y optimización se hace a través de un *controller*.

Finalmente, el *Pricing Core* provee su propia vista o interfaz, que establece su comunicación con el exterior. Se recomienda implementar este módulo mediante *Web Services*, reduciendo el acoplamiento con la interfaz de usuario, permitiendo incluso usar distintas plataformas para desplegar la aplicación.

3.3. Interfaz de Usuario

En el diseño de la interfaz de usuario es necesario hacer una correspondencia entre lo que se desea y lo que se debe mostrar, es decir, cómo organizar el contenido de la aplicación de modo que sea fácil de usar y se adapte al proceso de toma de decisiones de los usuarios (Nerukar, 2002). Para lograr este objetivo, se deben realizar cuatro tareas: (1) analizar el proceso de toma de decisiones, (2) determinar los objetos comunes de la interfaz, (3) establecer los módulos del sistema, y (4) modelar su flujo de forma que se completen satisfactoriamente las tareas dadas en (1).

3.4. El proceso de toma de decisiones

En este análisis se consideraron dos tipos de usuarios: (1) el Ejecutivo (o *Product Manager*), encargado de tomar la decisión de precio e inventario, y (2) el Gerente, quien supervisa la labor de los Ejecutivos y fija las reglas de negocio.

Comenzando con los Ejecutivos, se discute el análisis de productos. A pesar de que existen otras tareas (e.g. análisis de locales o de promociones) es en ésta donde se aplican los modelos de RRM, además de ser la más relevante para ellos. En la Figura 3.3 se muestra el diagrama de caso de uso del proceso, que comienza con la especificación de las tres dimensiones. Para los productos es posible seleccionar un grupo adicional, que permita hacer comparaciones. El siguiente paso es diferenciar si el análisis se realiza antes, durante o después del horizonte de venta. En el caso que sea antes, el Ejecutivo visualizará las diferentes curvas de un producto (o grupo de ellos) y generará un pronóstico de ventas, que mediante la optimización usará para determinar la cantidad inicial de inventario a ordenar junto con una curva de precios. Durante el período de ventas, el usuario verá (1) cómo han evolucionado en el tiempo, (2) cuál es la predicción para el resto del horizonte o temporada con la política de precios actual, y (3) si es posible mejorar los ingresos mediante una nueva optimización de precios. En el caso de realizar el análisis después de la

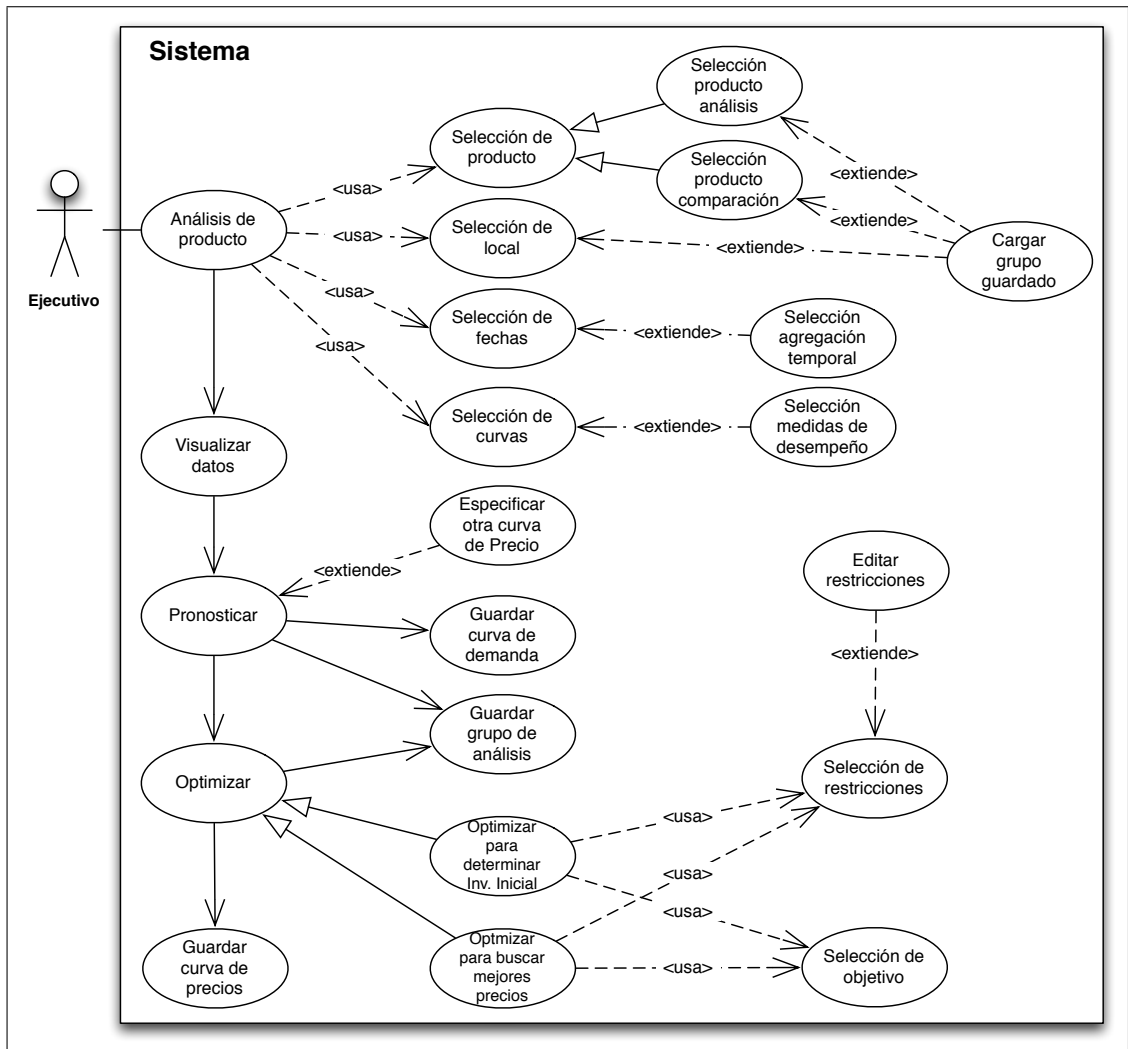


FIGURA 3.3. Proceso de análisis de productos.

temporada, se determina la precisión del pronóstico y se consideran otros escenarios de precio e inventario, con el correspondiente cambio en los beneficios.

El Gerente, tal como se mencionó, realiza dos labores principales: (1) supervisar la labor de los ejecutivos, lo que implica realizar sus mismas tareas pero a un alto nivel de agregación, y (2) fijar la reglas del negocio, cuyo diagrama de caso se uso de muestra en la Figura 3.4. El aspecto central es la edición de cada regla, que incluye la selección de los

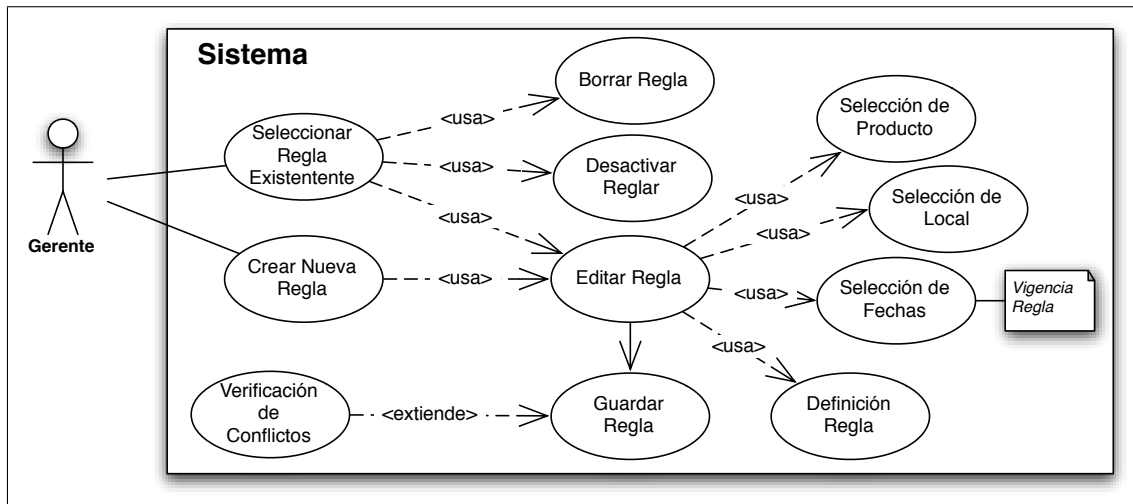


FIGURA 3.4. Proceso de fijación de reglas del negocio.

valores para las tres dimensiones y su definición, considerando los aspectos mencionados en la sección 2.2.

3.4.1. Objetos comunes de interfaz

Los objetos comunes de interfaz son aquellos que se repiten en los distintos módulos de la aplicación. Definirlos, a partir de las tareas descritas, entrega consistencia visual y funcional. Aunque su implementación específica dependerá de las tecnologías disponibles, es siempre necesario que permitan a los usuarios una interacción ágil y amigable. A continuación se describen seis objetos comunes de interfaz: tres relacionados con la selección de dimensiones, además de los gráficos, tablas y alertas.

Para que el usuario pueda establecer las dimensiones de análisis, es necesario un objeto *selector* de cada una. En el caso de las fechas, debe permitir especificar un rango y el nivel de agregación temporal. Para el caso de los productos y locales, debe posibilitar la selección de un conjunto de ellos, a partir de grupos definidos por la empresa o por los mismos usuarios. También debe agregar filtros propios del retailer, por ejemplo la temporada del producto en la industria de la ropa, o el formato del local en la industria de los comestibles.

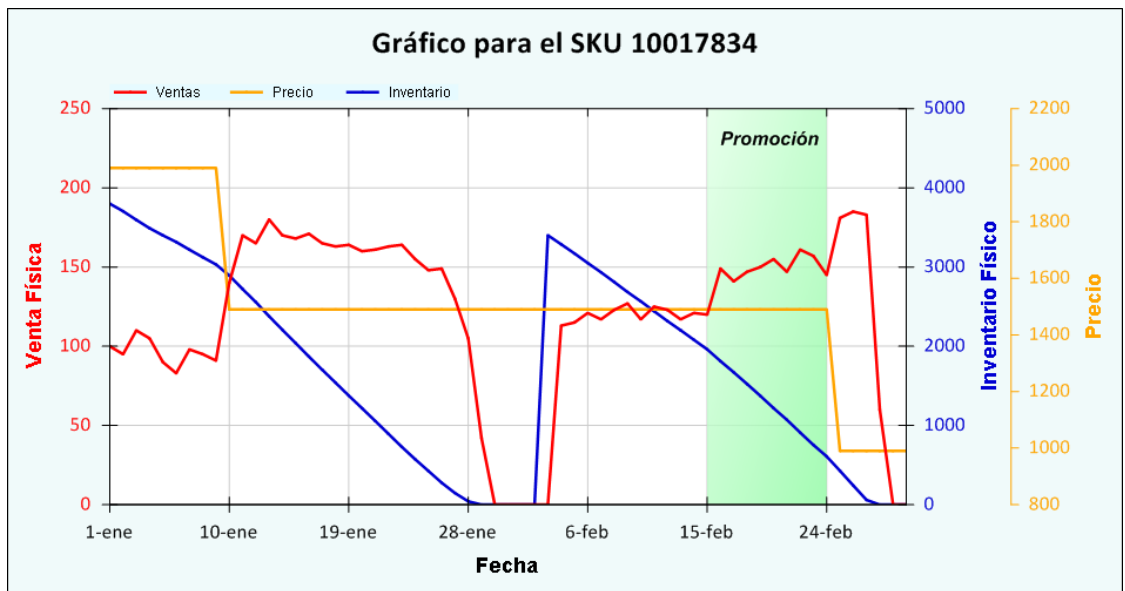


FIGURA 3.5. Ventas de un producto bajo efectos de precio, inventario y promociones.

Un cuarto objeto común son los gráficos. En la Figura 3.5 se muestra un ejemplo con la curva de ventas, inventario y precio de un producto. Las primeras aumentan a los 10 días debido a una baja del precio. En el día 25 se ven afectadas por los bajos niveles de inventario, hasta que finalmente se hacen cero. La región sombreada indica una promoción, que impacta positivamente las ventas del producto. Finalmente éste se liquida con el último cambio de precio. Cabe destacar que las nuevas tecnologías permiten gráficos cada vez más interactivos y dinámicos, lo que sin duda beneficia el análisis.

Los reportes por medio de tablas, el quinto objeto común, son útiles incluso cuando se dispone de una versión gráfica de la información. Ésta debe ser de fácil lectura y navegación.

Finalmente, las alertas para el usuario permiten advertir, por ejemplo, cuando una estrategia de precios no se ajusta a una determinada regla de negocio, o cuando una operación requiera una significativa cantidad de tiempo (adicionalmente, puede indicar el tiempo de ejecución aproximado).

3.4.2. Módulos de la Aplicación

En la implementación de la interfaz de usuario es recomendable utilizar una plataforma Web, ya que facilita la actualización, el manejo de los recursos computacionales y puede ser accedido desde casi cualquier lugar del mundo. Además, el avance actual de las tecnologías Web permite desarrollar aplicaciones con una flexibilidad y versatilidad similar a las de escritorio (véase Anexo A.4). La interfaz posee cinco módulos funcionales, que se describen a continuación: (1) administración, (2) visualización, (3) calibración, (4) optimización y (5) reportes.

Administración. Contiene las funciones de configuración propias del negocio. Difiere de la administración del sistema, puesto que está orientada a Ejecutivos y Gerentes, y posee cinco secciones:

- *Productos y Locales.* Muestra información relativa a productos, locales y sus agrupaciones. Adicionalmente, cada usuario puede definir sus propios grupos, según sus análisis y experiencia, lo que implica crearlos, editarlos y borrarlos.
- *Eventos y Promociones.* Permite definir el calendario de eventos y promociones. Si esta información no se encuentra estructurada ni sistematizada, el sistema pasa a ser un repositorio central. Una vez ingresada, debe ser posible su visualización y edición. También debe permitir ingresar simultáneamente un conjunto de promociones mediante un archivo de texto o planilla de cálculo, siguiendo un formato preestablecido. Esta característica resulta útil debido a que normalmente las promociones se planifican de forma anual y afectan a una gran cantidad de productos y locales. En la práctica resulta necesario asignar a un usuario específico que se encargue de mantener esta información actualizada.
- *Factores Exógenos.* Permite establecer estos factores y asignarles valores, cualitativos o numéricos.

- *Curvas de Demanda y Precio.* La aplicación de modelos de calibración y optimización darán paso a establecer distintas curvas de demanda y precio, asociadas a un grupo de productos y de locales. Debe ser posible administrar estas curvas, lo que implica crearlas, editarlas y borrarlas.
- *Reglas del negocio.* Esta sección permite a los Gerentes la administración de las reglas de negocio, siguiendo las consideraciones expuestas en la sección 3.4.

Visualización. Muestra a los usuarios la información disponible de manera gráfica y tabulada, constituyendo el primer acercamiento al análisis. Este módulo contiene los selectores de las tres dimensiones, un selector de curvas y un selector de promociones y eventos, y lo que entrega es un gráfico de las series en el tiempo con su respectiva tabla para un análisis más detallado.

Calibración. El objetivo de este módulo es generar pronósticos de demanda adecuados, que reflejen el comportamiento de los clientes al modificar la variable precio. Esto se logra mediante dos pasos: (1) ajuste del modelo y (2) pronóstico.

Para realizar el ajuste se requiere que el usuario seleccione las tres dimensiones (producto, local y fecha) y los factores o variables a incluir (discutidos en la sección 2.1). Una vez calibrado el modelo, se obtienen todos los estadísticos relevantes, tales como coeficiente, valor P, estadístico t, etc. El despliegue de esta información debe permitir al usuario ver claramente cómo afecta cada variable a la venta e incluir un gráfico que contraste la curva real con la calibrada. Además debe ser posible modificar esta última cambiando el valor de sus parámetros, agregando efectos no considerados por el ajuste. El modelo final puede ser guardado para análisis posteriores y para la optimización.

Una vez realizado el ajuste es posible generar pronósticos, para los mismos productos o para otros similares, por lo que es necesario especificar nuevas dimensiones. Lo que se obtiene es una comparación de las ventas pronosticadas con las reales (si existen), a través de un gráfico con su correspondiente tabla, además de medidas de error del pronóstico (si

corresponde). En esta sección es donde se produce el primer acercamiento del usuario con políticas de precios e inventario, ya que debe ser posible determinar el impacto que tiene en las ventas cambiar estos valores.

Optimización. El objetivo es determinar la trayectoria óptima de precios y la cantidad inicial de inventario a ordenar. Lo primero es proporcionar las dimensiones y un modelo de demanda, calibrado en la sección anterior. Con esta información se da paso a uno de los dos tipos de optimización, con o sin determinación del inventario inicial, asumiendo en este último caso un nivel fijo que puede proporcionar el usuario o bien basarse en la información del sistema. Un aspecto importante es la selección de restricciones que se usarán, incorporando las reglas de negocio establecidas en el módulo de administración. Una vez realizada la optimización no sólo se debe proveer la política óptima de precios (e inventario si corresponde), sino que también se debe dar al usuario la intuición detrás de la solución, ya que una mejor comprensión de ésta le permitirá comprender cómo influyen las distintas variables en su decisión. Esto se logra mediante gráficos y análisis de sensibilidad, que incluyan múltiples escenarios que se desvíen del original. Por ejemplo, es posible evaluar el impacto de agregar o quitar cierta restricción.

Reportes. Este módulo genera diversos reportes que no guardan relación directa con la aplicación de RRM, pero son de gran utilidad para la empresa y permiten obtener mayor provecho del sistema implementado.

Existen elementos generales que deben ser considerados en la interfaz. Un buen sistema de ayuda es vital en una aplicación de esta complejidad. Debe ser accesible en cada lugar donde puedan surgir dudas, y debe explicar no sólo las acciones posibles, sino también los modelos o procedimientos que hay detrás. Otro elemento importante consiste en contar con adecuadas herramientas de validación y manejo de errores, que sean fáciles de comprender por el usuario y que generen registros para ser consultados por los administradores del sistema.

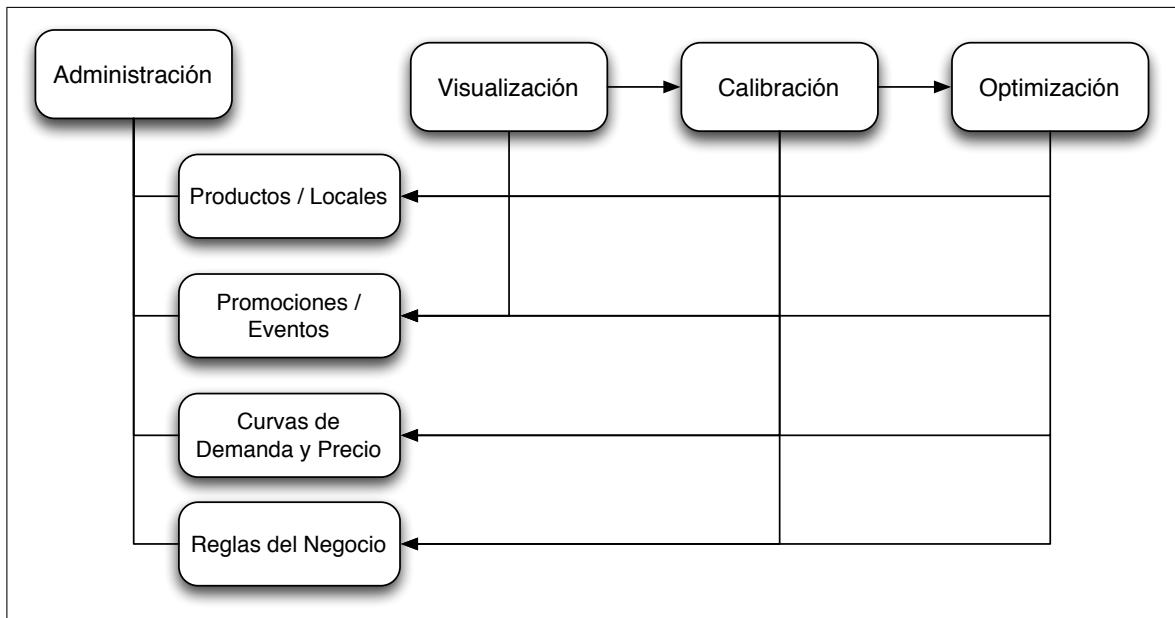


FIGURA 3.6. Flujo.

3.4.3. Flujo

En la Figura 3.6 se aprecia el flujo entre los módulos de interfaz descritos, establecidos a partir de los casos de uso presentados en la sección 3.4. Los distintos módulos pueden ser accedidos de forma independiente, pero también desde otros siguiendo las flechas de la figura. Por ejemplo, cada vez que se requiera seleccionar dimensiones puede ser necesario acceder al administrador de productos y locales, para manejar grupos de éstos definidos por el usuario. Algo similar ocurre con las otras secciones del módulo de administración.

El flujo comienza en la visualización, a partir de la cual el usuario puede pasar al módulo de calibración (con la misma especificación de dimensiones) para obtener una curva de demanda, o usar una que ya esté guardada. Luego se debe establecer una curva de precios que se ajuste al espacio factible dado por las reglas de negocio. Esto se logra de dos formas: (1) probando distintas políticas de precio y evaluando su impacto en el pronóstico, o (2) accediendo al módulo de optimización (con la misma especificación de dimensiones y curva de demanda).

4. CONCLUSIONES

Esta investigación, realizada en el marco de un proyecto de I&D en la industria del retail, proporciona un diseño para un DSS que reduce la brecha existente entre la teoría de Revenue Management y su aplicación a esta industria, permitiendo que una mayor cantidad de retailers puedan adoptar este sistema para determinar políticas óptimas de precio e inventario.

El sistema propuesto tiene la capacidad de incorporar diferentes modelos descritos en la literatura de RM usando una interfaz amigable que se va adaptando según el proceso de toma de decisiones de cada ejecutivo y/o empresa.

El presente diseño, exitosamente implementado en una cadena de retail chilena, se realizó a partir de proyectos de investigación aplicada en diversas empresas, por lo tanto, está hecho considerando problemas reales y relevantes de precio e inventario. La especificación de la base de datos, el Pricing Core y la interfaz de usuario provee una guía para la implementación del sistema, valiéndose de las tecnologías disponibles.

Las principales ventajas del uso del sistema propuesto son: (1) mejora el acceso a los datos debido a que los consolida, obteniendo incluso datos no sistematizados; (2) los presenta de un modo adecuado para el análisis; (3) permite al ejecutivo y/o empresa incorporar nuevas herramientas para un mejor análisis de la información, aplicando conocimientos de RM que, o no eran manejados, o era muy difícil de aplicar sin el apoyo de este DSS. Su uso no sólo agiliza el análisis, también mejora la comprensión de los distintos factores implicados en la decisión de precio e inventario; y (4) facilita la implementación de los precios obtenidos con el DSS mediante la interacción con los sistemas corporativos de la empresa.

Como extensión de esta investigación, se propone incorporar el diseño de un módulo para administrar la estrategia de precios, considerando las políticas y reglas establecidas por la empresa. Si se considera que cada ejecutivo debe manejar una alta variedad de productos, y que el tiempo es un recurso cada vez más escaso, sería productivo incorporar un

nuevo módulo que de manera automática determine y luego implemente políticas óptimas de precio sin necesidad de una constante intervención humana.

ANEXO A. GLOSARIO DE TÉRMINOS

A.1. GMROI

El GMROI o GMROIII (*Gross Margin Return on Inventory Investment*) es una medida de desempeño enfocada exclusivamente al retail, que indica cuántos dólares obtiene la empresa por cada dólar invertido de inventario, en un período de un año, permitiendo alinear las políticas de ventas con las de inventario. Se calcula de la siguiente manera:

$$\text{GMROI} = \frac{\text{Contribución Bruta}}{\text{Inventario Valorado Promedio}}$$

Donde:

$$\text{Contribución Bruta} = \text{Ingreso por ventas} - \text{Costo de los bienes vendidos}$$

Según Sweeney (1973), existen 5 ventajas en el uso de esta medida para el retail:

- (i) Es una medida de desempeño significativa para los ejecutivos, puesto que combina dos variables relevantes para ellos: el manejo de inventario de productos y la contribución.
- (ii) Es consistente con los objetivos financieros de la compañía.
- (iii) Dado una meta de GMROI, es posible establecer metas claras de contribución e inventario, definiendo distintas combinaciones que permitan obtener el GMROI deseado.
- (iv) Permite comparar varios departamentos o rubros del retailer, que pueden tener distintas características.
- (v) Se basa en datos que la empresa normalmente tiene disponible.

En el sitio Web de Retail Owners Institute (2008) se realiza una discusión más detallada del GMROI, sus usos y formas de calcularlo.

A.2. Object Relational Mapping

Se entiende por *Object Relational Mapping* (ORM) el proceso de transformar objetos de una aplicación a una representación que pueda ser persistida (almacenada) en una base de datos relacional (Fussell, 1997). Según Bauer y King (2005), el uso de estas herramientas de comunicación automatizada con la base de datos genera cuatro beneficios:

- (i) Aumenta la productividad, debido a reduce el tiempo tiempo de desarrollo de código para acceder a la base de datos.
- (ii) Genera menos líneas de código, y éste es más fácil de mantener, ya que se enfoca a resolver el problema de negocio.
- (iii) Integran una gran cantidad de optimizaciones, mejorando el rendimiento de la aplicación.
- (iv) Debido que una herramienta ORM abstrae la aplicación de la base de datos relacional, genera independencia del sistema administrador de base de datos que se utilice.

Actualmente existen diversas herramientas ORM, destacándose dos:

- Hibernate (<http://www.hibernate.org>)
- LINQ (<http://msdn.microsoft.com/en-us/data/cc299380.aspx>)

A.3. Decision Support Systems

Los denominados *Decision Support Systems* (DSS) son sistemas de información que buscan una mejora en la calidad de las decisiones. Esto se logra mediante la unión de de los recursos intelectuales de las personas con las capacidades computacionales de los sistemas, permitiendo enfrentar problemas semiestructurados o no estructurados (Keen y Morton, 1978). Los problemas no estructurados son aquellos que no poseen un algoritmo de solución, y se relacionan con la incerteza que puede existir para decidir entre varias opciones.

Estos sistemas hicieron su aparición en la década de 1960, cobrando mayor auge en la década de 1970. Poco a poco ganarían importancia, y adoptarían las tecnologías del momento: en los años 80 con la aparición de los computadores personales; en los años 90 con las arquitecturas cliente-servidor y los *Data Warehouse*; y a partir del año 2000 con las tecnologías Web. Hoy los DSS son cada vez más sofisticados, ocupando potentes herramientas de análisis, procesando gran cantidad de información, y facilitando el intercambio de conocimiento entre los distintos actores de las empresas.

Power (2002) clasifica los DSS en cinco tipos:

- (i) *Model-driven* DSS, que enfatizan el uso de modelos financieros, estadísticos, de optimización, de simulación, etc. Estos permiten a los usuarios analizar datos complejos, aunque no necesariamente requieren un uso intensivo de los datos.
- (ii) *Communication-driven* DSS, que permiten realizar tareas compartidas entre diferentes usuarios. Un ejemplo de estos sistemas es Microsoft Groove ¹.
- (iii) *Data-driven o data-oriented* DSS, tienen su foco en la obtención de series de tiempo provenientes de los datos de la compañía, almacenados en sus bases de datos o *DataWarehouses*.
- (iv) *Document-driven* DSS, permiten la búsqueda y manipulación de información contenida en documentos de diversos formatos.
- (v) *Knowledge-driven* DSS, que proveen sofisticados modelos de inferencia, basados en hechos, reglas, procedimientos y otras estructuras similares, almacenados en lo que se denomina “base de conocimiento”

¹<http://www.microsoft.com/latam/office/preview/programs/groove/overview.msp>

A.4. Tecnologías Web

El avance alcanzado por las tecnologías Web ha permitido la creación de aplicaciones con un nivel de interacción cada vez más parecido al que poseen las aplicaciones de escritorio. Hasta hace algunos años atrás, la interacción del usuario con una aplicación Web requería enviar los datos al servidor, para que este los procesara, y enviara la respuesta al usuario, en un proceso que requería la recarga de la página, y la consiguiente espera por parte del usuario. Esto hoy ha cambiado sustancialmente, lo que se ve reflejado en la aparición de la denominada Web 2.0, que mediante las RIA (Rich Internet Applications), mejora el intercambio de información y colaboración entre usuarios, además de la interacción con la aplicación.

Los *Web Services* han sido vitales en este desarrollo, cambiando la forma de crear aplicaciones. Estos permiten la interacción entre componentes de software, ubicados generalmente en diferentes computadores, sobre una red, normalmente Internet. Entre sus características, destacan (Hasan y Duran, 2006):

- Los *Web Services* son componentes autónomos, y su comunicación ocurre normalmente a través de mensajes XML bien definidos.
- Proveen una interfaz bien definida, mediante un documento XML llamado WSDL (*Web Services Description Language*), que entrega las operaciones que provee el servicio.
- Constituyen un punto de acceso para que consumidores y otros servicios se puedan suscribir, basados en la dirección y puerto del servicio (Típicamente una URL).
- Permiten crear piezas de software con bajo acoplamiento, menor que en el caso de los componentes orientados a objetos, generando una mayor flexibilidad. Un cambio en estos últimos hace necesario una nueva compilación, resultando en componentes fuertemente integrados a sus clientes o consumidores. Por otra parte, los consumidores de un servicio Web pueden responder flexiblemente a

cambios en la interfaz que provee el servicio, debido a que es fácil regenerar la conexión usando el documento WSDL actualizado, sin necesidad de volver a compilar.

La aparición de los Web Services ha dado origen a nuevas formas de hacer software. Gracias a los estándares establecidos por la W3C ² (donde el más usado es el SOAP), es posible crear aplicaciones sobre cualquier lenguaje y plataforma, que se comuniquen con otras desarrolladas en un lenguaje o plataforma diferente. Esto permite a los desarrolladores usar distintos servicios, que pueden estar en computadores de ubicación geográfica muy diferente. La idea planteada dio origen a la arquitectura SOA (*Service Oriented Architecture*), que representa una manera nueva y evolucionada para construir aplicaciones distribuidas. Los servicios son componentes distribuidos que proveen interfaces bien definidas y entregan mensajes XML. Esta arquitectura tiene por objetivo construir soluciones que vayan más allá de la organización, llegando fuera de los dominios de una empresa. Un negocio con muchos sistemas y aplicaciones en diferentes puede usar SOA para construir una solución integrada de bajo acoplamiento, que implemente flujos de trabajo unificados.

Esta nueva concepción de las aplicaciones ha cambiado incluso los modelos de negocios. Por ejemplo, Valente y Mitra (2007) muestran que los ASP (*Application Service Provider*) como modelo de negocio presentan una excesiva rigidez para las necesidades actuales, por lo que éste ha evolucionado hacia los e-Services. Valiéndose de Web Services, éstos presentan una orientación hacia los usuarios y clientes, ya que no se provee una aplicación propiamente tal, sino un servicio. Luego, la ésta se construye a partir de distintos servicios.

²World Wide Web consortium, <http://www.w3.org>

A.5. Business Intelligence

En la gestión de negocios, los principales problemas que se observan son la falta de rigurosidad en la aplicación del proceso de gestión, y la falta de información suficiente para tomar buenas decisiones.

Hoy en día, las empresas han implementado sistemas de información transaccional, encargados de recolectar los datos operacionales de la compañía. En muchos casos esta información se encuentra dispersa, y su volumen crece a un ritmo cada vez mayor. Esto genera entropía, vale decir el exceso de información no deja ver la que es relevante para la gestión del negocio.

La inteligencia de negocios (*Business Intelligence*, BI) aborda este problema, por medio de sistemas, procesos y tecnologías que recolectan, integran, analizan y presentan conocimiento e información relevantes para la gestión de la empresa. Están basados en el desarrollo de modelos complejos en distintas áreas de patrones de compra, servicio al cliente, desarrollo de productos, operaciones, finanzas, detección de fraude, predicción de fugas, etc. En Hancock y Toren (2005) es posible encontrar un detallado análisis y metodología de aplicación de BI.

Toda compañía de un tamaño razonablemente grande posee sistemas para el manejo de sus datos. Estos sistemas son usualmente los denominados OLTP (OnLine Transaction Processing), encargados de manejar miles de transacciones diarias. Su objetivo principal es proveer consistencia de la información y la habilidad de soportar adiciones y modificaciones de pequeñas cantidades de información cada vez. Estos sistemas se modelan usando un esquema Entidad-Relación (ER), que produce una estructura normalizada, donde cada entidad tiene su propia tabla separada, relacionada con otras. Esto funciona bien almacenando datos operacionales de la empresa, pero falla a la hora de generar reportes. Cualquier consulta al esquema OLTP implica una buena cantidad de tablas, unidas por medio de *joins* y filtros. El tiempo de ejecución de esas consultas probablemente no

será adecuado para una base de datos de tamaño medio, independiente del software o hardware que se utilice.

Para solucionar este problema, aparecieron los *Data Warehouses* (DW). Estos son almacenes de datos que muestran una vista consistente de ellos, y facilitan el acceso de los usuarios a la información, ya que presenta los datos de modo que pueda responder a las preguntas de gestión que se realizan los usuarios. Estos obtienen sus datos de las bases de datos operacionales de la empresa mediante procesos ETL (*Extraction, Transformation and Loading*). La clave del alto desempeño de un DW es la tecnología OLAP (*OnLine Analytical Processing*), diseñada específicamente para BI. Estas son bases de datos modeladas dimensionalmente.

La modelación dimensional comienza con la definición de sus medidas, que son los indicadores de negocio que se necesitan conocer. En un retailer, estos pueden ser los ingresos, el margen, la contribución, etc. Con esto, se crean las *Fact Tables*, que contienen las medidas que se definieron previamente. Ahora bien, estas medidas pueden ser vistas de distintas maneras. Cada una de estas formas es conocida como *dimensión*. Ejemplos de dimensiones son producto, local, o tiempo. Cada dimensión genera su propia tabla, asociada a una *Fact Table*. Esto genera un diseño de tipo *Star*, o *Snowflake* si es que existen jerarquías en las dimensiones.

Con este modelo de DW, se genera un proceso ETL que calcula los valores de las medidas, poblando las *Fact Tables*, lo que permite presentar los datos al usuario de un modo adecuado para su gestión, con una velocidad de respuesta mucho mayor respecto a una base de datos relacional. Esta estrategia sin duda aumenta la capacidad de análisis de los ejecutivos, lo que implica mejores decisiones. También sienta las bases para otras metodologías, como la minería de datos.

BIBLIOGRAFIA

- Akaike, H. (1974, Diciembre). A new look at the statistical model identification. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, 19(6), 716-723.
- Anderson, E., y Simester, D. (2003). Effects of 9 price endings on retail sales: Evidence from field experiments. *Quantitative Marketing and Economics*, 1(1), 93-110.
- Arnott, D., y Pervan, G. (2005, Abril). A critical analysis of decision support systems research. *Journal of Information Technology*, 20(1), 67-87.
- Arnott, D., y Pervan, G. (2008, Febrero). Eighth key issues for the decision support systems discipline. *Decision Support Systems*, 44(3), 657-672.
- Aviv, Y., y Pazgal, A. (2005, Septiembre). A partially observed markov decision process for dynamic pricing. *Management Science*, 51(9), 1400-1416.
- Bauer, C., y King, G. (2005). *Hibernate in action*. Manning Publications.
- Baumgartner, R., Frölich, O., Gottlob, G., Harz, P., Herzog, M., y Lehmann, P. (2005). *Web data extraction for business intelligence: the lixto approach*.
- Bertsimas, D., y Perakis, G. (2006). Dynamic pricing: A learning approach. *Mathematical and Computational Models for Congestion Charging*, 101(10), 45-80.
- Bhagarva, H., Power, D., y Sun, D. (2007, Agosto). Progress in web-based decision support technologies. *Decision Support Systems*, 43(4), 1083-1095.
- Bitran, G., y Caldentey, R. (2003). An overview of pricing models for revenue management. *Manufacturing & Service Operations Management*, 5(3), 203-229.
- Bitran, G., y Mondschein, S. (1997). Periodic pricing of seasonal products in retailing. *Management Science*, 43(1), 64-79.

- Cassaigne, N., y Singh, M. G. (2001, Febrero). Intelligent decision support for the pricing of products and services in competitive consumer markets. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, 31(1), 96-106.
- Elmaghraby, W., y Keskinocak, P. (2003, Octubre). Dynamic pricing in the presence of inventory considerations: Research overview, current practices, and future directions. *Management Science*, 49(10), 1287-1309.
- Friend, S., y Walker, P. (2001, November). Welcome to the new world of merchandising. *Harvard Business Review*.
- Fussell, M. (1997). *Foundations of object relational mapping*.
- Gallego, G., y Van Ryzin, G. (1994, Agosto). Optimal dynamic pricing of inventories with stochastic demand over finite horizons. *Management Science*, 40(8), 999-1020.
- Hancock, J., y Toren, R. (2005). *Practical business intelligence with sql server 2005*. Addison Wesley Professional.
- Hasan, J., y Duran, M. (2006). *Experte service-oriented architecture in c# 2005* (Second ed.). Springer-Verlag.
- Heching, A., Gallego, G., y Van Ryzin, G. (2002, April). Mark-down pricing: An empirical analysis of policies and revenue potential at one apparel retailer. *Journal of Revenue and Pricing Management*, 1(2), 139-160.
- Johnson, C. (2001). *Retail revenue optimization: Time and rewarding* (Tech. Rep.). Forrester Research Inc.
- Jukié, N. (2006, Abril). Modeling strategies and alternatives for data warehousing projects. *Communications of ACM*, 49(4), 83-88.
- Keen, P., y Morton, M. S. (1978). *Decision support systems: An organizational perspective*. Addison Wesley.
- Kincaid, W. M., y Darling, D. A. (1963). An inventory pricing problem. *Journal of Mathematical Analysis and Applications*, 7, 183-208.

- Mantrala, M. K., y Rao, S. (2001, Mayo-Junio). A decision-support system that helps retailers decide order quantities and markdown for fashion goods. *Interfaces*, 31(3), 146-165.
- Montgomery, A. (1997, Abril). Creating micro-marketing pricing strategies using supermarket scanner data. *Marketing Science*, 16(4), 315-337.
- Montgomery, A. (2005, Julio-Octubre). The implementation challenge of pricing decision support systems for retail managers. *Applied Stochastic Models in Business and Industry*, 21(4-5), 367-378.
- Nerukar, U. (2002). Web user interface design: Forgotten lessons. *IEEE Software*, 18(6), 69-71.
- Power, D. (2002). *Decision support systems: Concepts and resources for managers* (DSSResources.COM, Ed.). Quorum Books division Greenwood Publishing.
- Retail Owners Institute. (2008). *Core j2ee patterns - data access object*.
- Smith, S., y Achabal, D. (1998). Clearance pricing and inventory policies for retail chains. *Management Science*, 44(3), 285-300.
- Smith, S., McIntyre, S., y Achabal, D. (1994, Febrero). A two-stage sales forecasting procedure using discounted least squares. *Journal of Marketing Research*, 31(1), 44-56.
- Sun Microsystems Inc. (2002). *Core j2ee patterns - data access object*.
- Sweeney, D. (1973, Enero). Improving the profitability of retail merchandising decisions. *Journal of Marketing*, 37(1), 60-68.
- Talluri, K., y Van Ryzin, G. (2004). *The theory and practice of revenue management* (S. S. . B. M. Inc., Ed.). Kluwer Academic Publishers.
- Valente, P., y Mitra, G. (2007, 1116). The evolution of web-based optimisation: From asp to e-services. *Decision Support Systems*, 43(4), 1096.
- Valkov, T. (2005). From theory to practice: Real-world applications of scientific pricing across different industries. *Journal of Revenue and Pricing Management*, 5(2), 143-151.

Van Ryzin, G. (2005, April). Future of revenue management: Models of demand.
Journal of Revenue and Pricing Management, 4(2), 204-210.