

Integración de Modelos de Planeamiento y *Scheduling* en Sistemas de Información Empresariales

Melina C. Vidoni¹, María Analía Rodríguez², Aldo R. Vecchietti³

^{1,2,3} INGAR-CONICET, UTN FRSF, Santa Fe, Santa Fe, Argentina
{melinavidoni, r_analia, aldovec}@santafe-conicet.gov.ar

Abstract-- La aplicación e implementación de modelos matemáticos tendientes a colaborar con la toma de decisiones no resulta un proceso sencillo y directo: existen diversos factores que dificultan la integración de los modelos matemáticos desarrollados con los sistemas de información empresariales. ERP 2 Optimizer Linkage (E2OL) es un sistema que acopla un modelo matemático al sistema empresarial o ERP utilizado, proveyendo las interfaces necesarias para la adecuada visualización y facilitando el modelado de problemas típicos presentes en la gestión de empresas de producción, al incrementar el empleo de los modelos de optimización de la planificación y scheduling en los procesos de manufactura de las empresas.

Keywords-- ERP, optimización, modelos matemáticos, planning, scheduling, integración.

1. Introducción

En la actualidad, la injerencia de los sistemas de información (SI) se ve reflejada en todas las actividades centrales de las empresas, cobrando un rol de mayor preponderancia en los procesos de negocios que la misma lleva a cabo. En este sentido, se plantea una nueva relación entre las organizaciones y los sistemas de información, éstos desempeñan sin duda un rol estratégico en la vida de la compañía (Laudon & Laudon, 2004).

Sin embargo, la incorporación de costosos SI, como por ejemplo los ERP (*Enterprise Resource Planning*), ha traído resultados no siempre positivos, debido al costo y esfuerzo de implementación, como así también los profundos cambios organizacionales que trae. Un problema es que los ERP no proveen herramientas avanzadas de optimización que permitan la toma de decisión en situaciones complejas y específicas. Si bien existen numerosos modelos de planeamiento y scheduling muy pocos de estos se han llevado a la práctica y se han implementado en los sistemas de información empresariales, produciendo una brecha entre la teoría y la práctica en estos sistemas (Framinan & Ruiz, 2010). Esto hace que muchas empresas deban desarrollar sus propios modelos de optimización y las interfaces para integrarlos con los ERP, restándole propiedades deseables de todo sistema, como son su fácil modificación y mantenimiento, la capacidad de soportar el crecimiento de la empresa, etc.

La planificación y el *scheduling* son procesos de toma de decisiones usados regularmente dentro de las industrias de manufacturas y servicios, importantes en diversas

áreas (producción, inventario, transporte, distribución, etc.), los cuales pueden resolverse usando técnicas basadas en reglas, heurísticas y modelos de optimización que facilitan su resolución en tiempos sensatos (Pinedo, 2005). Como procesos de toma de decisiones, tienen un rol muy importante en la manufactura y en los sistemas de producción; esto incluye los ambientes de procesamiento de información (Pinedo, 2012).

Considerando esta problemática, se han desarrollado varios trabajos en el área. Bourgeois, Artiba y Tahon (1993) presentan un sistema de scheduling a corto plazo desarrollado específicamente para una empresa. Por su parte, Lee, Jeon y Moon (2002) presentan un modelo para un sistema APS (*Advanced Planning and Scheduling*) basado en algoritmos genéticos, que trabaja con órdenes de venta con fecha de entrega y que permite considerar tercerización. En Lala y otros (2005) se propuso un modelo combinatorio, posteriormente implementado en C++, para integrar sistemas MRP (*Material Requirements Planning*) con *scheduling* tipo shop-floor en ambientes discretos. Los autores Masuchun, Masuchun y Thepmanee (2009) desarrollaron un modelo para integrar el *scheduling* m-máquina en sistemas MRP, pero no llegaron a implementar un sistema que emplee dicho modelo. Otro trabajo relacionado es el de Van Nieuwanhuysen y otros (2011), donde los autores desarrollaron un módulo ARP (*Advanced Resource Planning*) el cual provee un proceso de selección de parámetros con la meta de proveer información realista para el *scheduling*; también implementaron este módulo en varias empresas. Finalmente, Hvolby y Steger-Jensen (2010) realizaron un análisis sobre los nuevos sistemas APS, concluyendo que son los sistemas que pueden ofrecer funcionalidades avanzadas de planeamiento y *scheduling*, enlazándose con los ERP.

Si bien se ha producido mucha investigación en el área, algunas de las cuales han llegado a implementarse, la gran mayoría de las propuestas tienen un problema: se trata de módulos APS específicos para una sola empresa, o módulos planteados como “genéricos” que si bien tienen características avanzadas respecto a los ERP, utilizan modelos matemáticos (o algoritmos) genéricos que no permiten considerar cuestiones particulares del negocio de la empresa donde se va a implementar.

Tomando esto como premisa, se desarrolló un sistema de software que enlaza el sistema ERP de una empresa -o el sistema propio ya existente- con un modelo matemático de optimización (el cual puede ser reemplazado y/o modificado), para poder ejecutar el *planning* y *scheduling* de la producción integrado al ERP. Dicho sistema fue nombrado E2OL (siglas en inglés de *Enterprise Resource Planning To Optimizer Linkage*).

2. Diseño de E2OL

El desarrollo de E2OL fue guiado por el planteo de requerimientos funcionales.

Inicialmente, se definió que el sistema debía requerir la autenticación del usuario con las mismas credenciales que las utilizadas en el ERP. Las demás funcionalidades sólo estarían disponibles al acceder exitosamente. E2OL debía permitir listar los órdenes/pronósticos de ventas (según lo utilizado por la empresa) en una tabla de campos personalizables, para que el usuario elija de ellas las entradas de datos al modelo

matemático. Dicho modelo debe ser reemplazable y no estar ligado al sistema. Una vez finalizada la ejecución del modelo, la cual debe ser transparente para el usuario, los resultados deben mostrarse de forma gráfica, en tablas o en el formato que el planificador seleccione. El administrador debe poder configurar el modo de presentar los resultados y el contenido de los mismos. El planificador puede guardar los resultados del *scheduling* en el ERP o ejecutar el modelo matemático con nuevos escenarios hasta que los resultados obtenidos lo satisfagan. Por otro lado, los atributos de calidad del software son factores que afectan el diseño del sistema, y su posterior comportamiento en tiempo de ejecución. Teniendo en cuenta la importancia de fijar atributos de calidad y considerando la problemática que se quiere resolver, se definieron varios atributos de calidad importantes, que guiaron el desarrollo del sistema:

- **Modificabilidad:** que es la habilidad para realizar cambios en el sistema de manera rápida y con un mínimo de impacto (a bajo costo). E2OL debe ser adecuado a cada organización y al sistema de información que ésta posea, configurando la base de datos con la que se conecta, la forma de obtención de los datos del modelo y la forma de presentar los resultados. Además debe poder adaptarse fácilmente y con un costo mínimo a los cambios de requerimientos futuros.
- **Portabilidad:** habilidad de un sistema de correr bajo diferentes entornos, los cuales pueden ser hardware, software, o una combinación de ambos. E2OL debe ser fácilmente configurable para adaptarse a los diferentes ambientes de trabajo y a las necesidades de cada una de las empresas. Los componentes que se emplearon para la arquitectura de E2OL permiten una fácil adaptación y portabilidad.
- **Interoperabilidad:** es la propiedad que dos o más sistemas cooperen entre ellos en tiempo de ejecución. Este atributo se cumple en E2OL porque la arquitectura que se propone utiliza la misma tecnología de base de datos que el sistema donde se hospeda, pudiendo cooperar de manera natural, obtener datos y guardar resultados en el sistema empresarial.
- **Usabilidad:** es la facilidad de uso y de entrenamiento de los usuarios del sistema; debe ser de fácil aprendizaje, eficiente y controlable. El funcionamiento interno de E2OL debe ser transparente para el usuario, requiriendo mínimos conocimientos de su entorno y como se ejecutan. La curva de aprendizaje no debe ser elevada ni prolongada en el tiempo. La adaptación al entorno de trabajo debe ser sencilla. En este atributo de calidad se puso especial énfasis, ya que uno de los objetivos para el desarrollo de E2OL es facilitar el uso de las herramientas de optimización de la producción en los sistemas empresariales; para lograr esto el usuario debe contar con una herramienta que sea simple de usar.

2.1 Arquitectura de E2OL

Los ERP son sistemas de software desarrollados en módulos, donde cada uno responde a funcionalidades particulares, y son implementados módulo-a-módulo (Muscatello, Small, & Chen, 2003). La arquitectura más clásica de estos sistemas, adoptada por los principales proveedores de ERP propietarios, es cliente/servidor

(Montagna & Leone, 2000).

Por lo mencionado previamente, y debido a su utilización en el desarrollo de sistemas ERP, se decidió que la estructura básica del sistema, sería tipo cliente/servidor.

Tomando esto como base, el lado cliente de E2OL fue estructurado siguiendo los módulos detallados por Pinedo (2005), para aquellos sistemas que integran los procesos de planeamiento y “scheduling” con sistemas empresariales: (a) módulo de bases de datos o conocimiento, (b) módulo dedicado al *solver* y (c) módulo de interfaces gráficas.

Finalmente, se analizó un conjunto de datos que no pueden estar embebidos en el código, ya que de esa manera no podrían garantizarse los atributos de portabilidad y modificabilidad. Esta información es fundamental para llevar a cabo la ejecución de los modelos de planeamiento y *scheduling*, a saber:

- a. Información base de datos del ERP (BDERP) o del sistema empresarial, como ser: datos de conexión, estructura de las tablas, atributos de la misma, etc. También se incluyen las sentencias SQL necesarias (de selección o inserción de datos) para acceder y/o guardar información en esas tablas.
- b. Datos relacionados con el modelo matemático de optimización, el archivo del modelo mismo, los datos de entrada y su origen. los resultados/errores de la ejecución del modelo.
- c. Configuración de la presentación de resultados, incluyendo campos, formatos y contenidos a mostrar.

La arquitectura que se propuso para E2OL se centró en la creación de una segunda base de datos, denominada Nomenclaturas (BDN), la cual gestiona esta información relevante, manteniéndola aislada del código del sistema, y favoreciendo la independencia de los datos. Al poseer una estructura estándar, la BDN puede implementarse en el mismo Sistema de Gestión de Bases de Datos del sistema empresarial. La Figura 1 muestra el diagrama completo de la arquitectura del sistema.

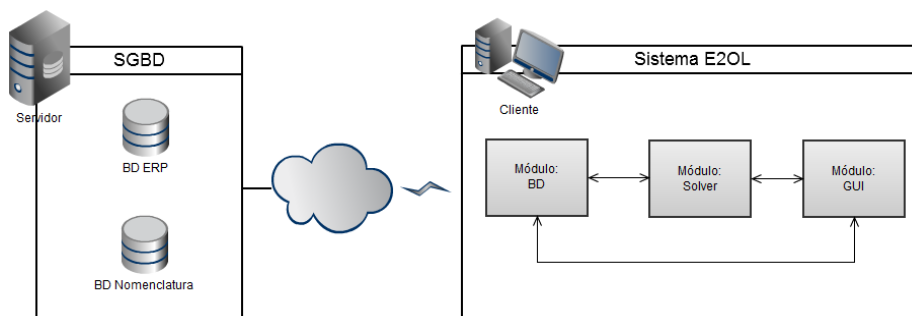


Figura 1. Arquitectura completa del sistema E2OL.

2.2 Herramientas Utilizadas

Para favorecer la modularización del sistema, la reutilización del código y la portabilidad del sistema, se encaminó el desarrollo de E2OL a un lenguaje de

programación orientado a objetos, independiente de la plataforma donde se ejecute y ampliamente conocido. Por esto mismo, se seleccionó Java.

Java es un lenguaje de programación orientado a objetos, de propósito general, concurrente y basado en clases, específicamente diseñado para tener la menor cantidad posible de dependencias de implementación. En este lenguaje, el código que corre sobre una plataforma no necesita ser recompilado para correr en otra (Oracle, 1999).

3. Funcionamiento del *Solver*

El optimizador matemático funciona desde el lado cliente del servidor empleando, para el ejemplo que se presenta en este artículo, *General Algebraic Modeling System* (GAMS), el cual es un sistema de modelado de alto nivel para programación y optimización matemática. Consiste de un compilador del lenguaje y varios *solvers* de elevada performance integrados al mismo. El API de Java provee los métodos y elementos necesarios para ejecutar un modelo matemático desde un sistema Java, enlazando éste con el *solver* adecuado (Dirkse & Bussiek, 2011). La arquitectura propuesta puede incluir la ejecución de cualquier sistema de programación y modelado distinto a GAMS, se empleo este último porque se cuenta con una licencia educativa.

GAMS se encuentra ubicado dentro del módulo del lado cliente, dedicado a la ejecución del *solver*. Además de esto, dicho módulo posee clases y paquetes abocados a la traducción de los datos de entrada al modelo, y a traducir los resultados del mismo hacia los gráficos predeterminados.

En las siguientes subsecciones se detalla más a fondo el funcionamiento de este módulo.

3.1 Ejecución del Modelo

El proceso de ejecución del modelo matemático puede verse en el diagrama de actividad de la Figura 2:

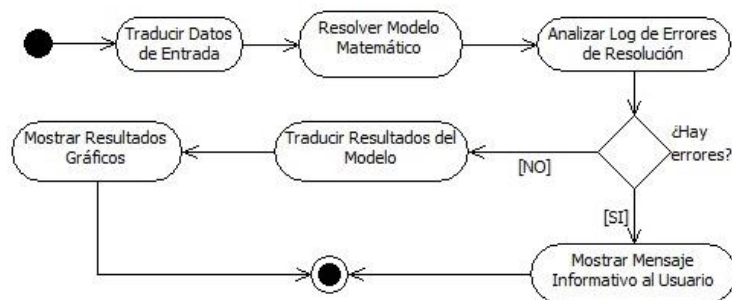


Figura 2. Diagrama de actividad del proceso de ejecución del modelo matemático.

La actividad *Traducir Datos de Entrada* implica obtener los datos de entrada del modelo matemático, y escribirlos en archivo de texto con el formato en que los necesita el mismo. Hay dos tipos de datos de entrada, los cuales junto al proceso, serán descriptos

con mayor detalle en la sub-sección 3.2.

A continuación, *Resolver Modelo Matemático* ejecuta GAMS para obtener una solución al modelo, empleando los datos de entrada generados en la actividad anterior. En este paso, el sistema exige que el modelo genere un log de errores, el cual es posteriormente analizado en la actividad *Analizar Log de Errores*. En este punto, el sistema evalúa si el código escrito corresponde o no a una situación de error; en caso positivo, se muestra un mensaje al usuario, informando del mismo. La tabla 1 presenta ejemplos de casos exitosos o no exitosos:

Casos Exitosos	Casos No Exitosos
Óptimo Global	Infactible
Óptimo Local	No Acotado
Solución Entera	No Entera

Tabla 1. Ejemplo de casos exitosos y no exitosos en la resolución de un modelo matemático.

Por otro lado, ante un caso exitoso de resolución, la actividad *Traducir Resultados del Modelo* es la encargada de traducir los resultados del modelo matemático hacia los resultados gráficos configurando, no sólo el tipo de gráfico, sino también el contenido del mismo. Cuando esto termina, el sistema muestra todos los resultados al usuario, mediante la actividad *Mostrar Resultados Gráficos*.

Todo el proceso detallado previamente puede tomar varios segundos para la ejecución, lo que principalmente depende del tipo de modelo y de las restricciones que el mismo posea, la cantidad de datos de entrada, la complejidad del *scheduling* a realizar, la capacidad de procesamiento del hardware, entre otras cosas.

Sin embargo, toda esa complejidad es transparente al usuario quien, durante este proceso, observa por pantalla un diálogo con una barra de progreso indeterminada (Fig. 3), cuyo mensaje se actualiza con cada avance en el proceso.

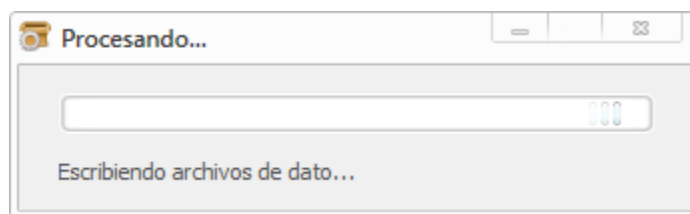


Figura 3. Diálogo con barra de progreso, visualizado por el usuario.

3.2 Traducción de Datos de Entrada

Por motivos de simplicidad para el manejo de los archivos de entrada, los mismos fueron separados en dos categorías, para facilitar la implementación: (a) *archivos tipo*

datos, los cuales son cargados con datos extraídos del ERP o sistema empresarial al cual se ligue E2OL y (b) *archivos tipo parámetros*, que son cargados con valores que el usuario ingresa mediante una interfaz gráfica, y que le permiten analizar diferentes escenarios para un mismo conjunto de dato.

Ambos tipos de archivos conforman el conjunto de todos los datos de entrada, pero su uso y composición es diferente. De esta forma, el proceso de traducción de los datos de entrada implica obtener la configuración de los archivos tipo (a) y luego cargar desde la BDERP los archivos tipo (b).

El proceso para la carga de los primeros, puede verse a continuación, en el diagrama de actividad de la Figura 4:

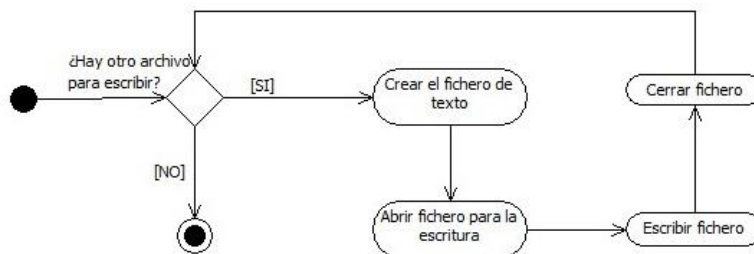


Figura 4. Proceso para la traducción de los archivos tipo parámetro.

Como puede observarse, el proceso se repite por cada uno de los archivos a configurar.

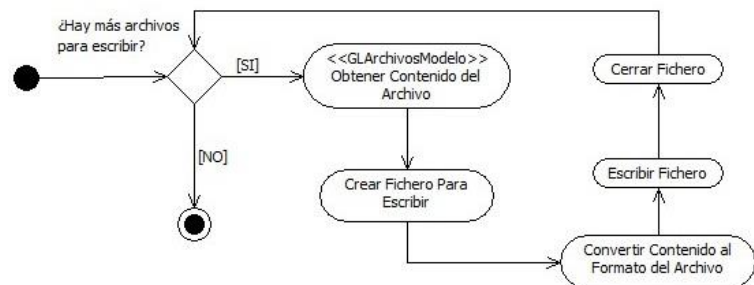


Figura 5. Proceso para la traducción de archivos de entrada tipo dato.

Sin embargo, la traducción presenta variaciones al momento de escribir los archivos tipo datos. Siguiendo el diagrama de actividad de la Figura 5, se observa la adición de una nueva actividad denominada *Obtener Contenido del Archivo*, la cual implica conectarse a la base de datos de Nomenclaturas (BDN), obtener los datos de las consultas SQL a ejecutar sobre la BDERP, efectuar la consulta y luego traducir los datos al formato que GAMS solicita. Nuevamente, el proceso de la Fig. 5 se realiza para cada archivo.

3.3 Traducción de Datos de Salida

El proceso inicia solicitándole a un gestor del módulo de BD, la información de dónde

obtener los datos de los resultados, los cuales son devueltos en una lista de objetos. Los tipos de gráficos disponibles, son: diagrama Gantt, tablas de entrada simple (cabeceras sólo en las columnas) y entrada doble (cabeceras en las filas y en las columnas), gráfico de barras, gráfico de tortas y diagrama de puntos. La Figura 6 muestra un ejemplo de la interfaz con los gráficos.

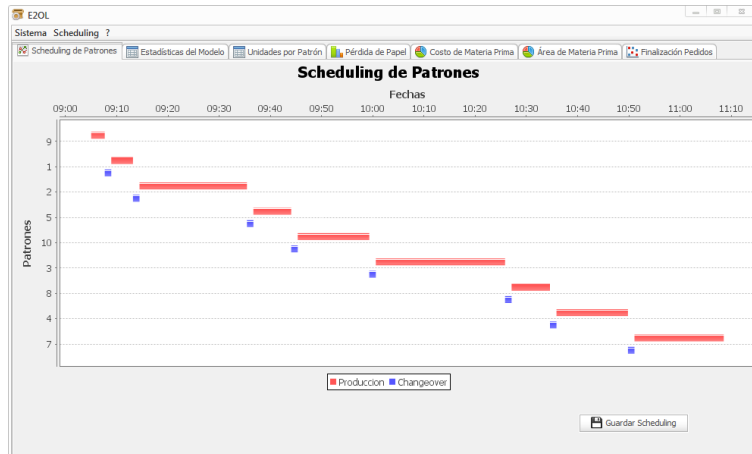


Figura 6. GUI con resultados gráficos del modelo matemático.

El siguiente paso, es efectivamente leer los datos necesarios del archivo de resultados señalado, ubicando la región del texto del mismo que efectivamente contiene la información de cada resultado. Para esto, y por cada uno, busca en el archivo de resultados la línea que contiene el texto de inicio que precede a los datos necesarios

Finalmente, prepara el panel donde se cargarán los datos, el cual es agregado a un listado de paneles que el método arma, y que devuelve al hilo en ejecución.

4. Caso de Estudio

Se planteó un caso de estudio que permitiera evaluar al sistema en una situación real y acotada. Para esto mismo, se utilizó un modelo de empresa mediana, llamada *Todo Cartón*, la cual es una empresa manufacturera mediana, productora de cajas de cartón corrugado personalizadas y bajo pedidos. De este modo, el proceso de producción es planificado para lograr cubrir los requerimientos de la caja, considerando cuestiones de producto (como ser las dimensiones, tipos de papeles a emplear, etc.) y de planificación (es decir, cantidad de unidades requeridas, y fechas de entrega). Esto mismo hace que sea una necesidad importante poder mantener una gestión integrada de las órdenes de venta, de los pedidos, producción y planificación.

4.1 Funcionamiento de *Todo Cartón*

La empresa mantiene en inventario rollos de papel que compra a sus proveedores, y que utiliza para la producción de las cajas. Estos papeles son de diferentes gramajes, colores, y anchos; esto último para poder administrar la producción y lograr el menor desperdicio posible. La caja se produce empleando varias capas de papel, donde algunas son lisas y otras ondeadas otorgándole la rigidez requerida.

Por otro lado, también se consideran las dimensiones de la caja en el momento de la realización del pedido para diseñar un molde de corte para la misma (plancha), que el cliente debe aprobar (ver Figura 7). Las planchas de diferentes dimensiones (y proveniente incluso de pedidos de diversos clientes) pero de un mismo tipo de cartón (igual cantidad de capas de papel, tipos de papeles en cada una y tipos de onda) pueden ser producidas simultáneamente en la máquina de corte a partir de lo que se denomina un patrón de corte.

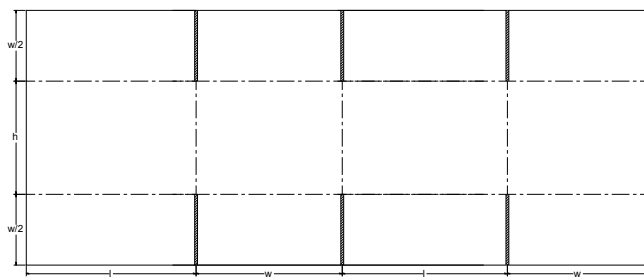


Figura 7. Ejemplo de patrón de corte de una caja de cartón.

El proceso de producción implica preparar los papeles en la máquina, alinear las cuchillas y guillotinas que irán haciendo los cortes, y luego procesar el papel (lo que implica ondear la capa intermedia y pegar las tres capas con pegamento); una vez hecho esto se general los cortes longitudinales y luego los transversales.

Por otro lado, la gestión de la información dentro de *Todo Cartón* se realiza mediante un sistema empresarial adaptado para cubrir sus necesidades. Dicho sistema es *OpenERP* (Pinckaers, Gardiner, & Van Vossel, 2011), un ERP de código abierto e implementación gratuita, desarrollado en PHP y con base de datos en *PosgreSQL*.

4.2 Modelo Matemático de *Todo Cartón*

Con los detalles de la empresa mencionados anteriormente, se reconoce que *Todo Cartón* encuentra la necesidad de implementar un modelo de optimización matemática que resuelva el problema de planificación y *scheduling* para la producción de cajas de cartón. Para esto, se decide implementar a E2OL en la firma, e integrarlo con el actual sistema empresarial, *OpenERP*.

El modelo de optimización matemática implementado fue desarrollado en (Rodríguez & Vecchietti, 2013), como parte de un proceso de mejora de la empresa de la que fueron tomados los datos para simular a *Todo Cartón*. Este modelo busca minimizar la pérdida de

cartón, dado que significa un alto costo para la empresa. La empresa mantiene en stock papeles con diferentes anchos a fin de contar con cierta flexibilidad para realizar la planificación. A su vez, se cuenta la cantidad de cajas mínimas requeridas por pedido y una cota de sobreproducción para satisfacer la demanda. En la formulación inicial, las restricciones de demandan dan origen a desigualdades con términos bi-lineares. Términos de éste tipo también se presentan en las restricciones de asignación de papeles a patrones de corte, y en las de cálculo de área. El modelo es luego reformulado como un Mixto Entero Lineal.

Por otro lado, el *scheduling*, integrado al modelo de planificación de corte, consiste en definir el orden en que los patrones de corte deben ser producidos. Dado que los tiempos de procesamiento no son conocidos previamente sino que dependen del plan de corte, una representación continua es más apropiada para el problema.

Los productos a manufacturar no son secuenciados de forma directa, ya que son asignados a los patrones, como planchas de cartón. La cantidad de unidades solicitadas para cada uno se satisface usando uno o más patrones de corte, así que el tiempo de finalización de un producto considera todos los patrones en los que éste es asignado. De esta manera, la orden termina cuando concluye la producción del último patrón de corte al cual está asignado.

4.2.1 Resultados Gráficos Solicitados

Para el caso de estudio, los resultados solicitaron eran, principalmente, pertenecientes a tres categorías: (a) información pertinente a la configuración de los patrones (papeles, anchos, capas, etc.), (b) información de la producción (pedidos por patrón, diagrama de Gantt de la producción de patrones, y diagrama de puntos de la fecha de finalización de pedidos) y (c) análisis de relación materia prima utilizada/perdida, en distintos aspectos.

4.3 Utilización de E2OL

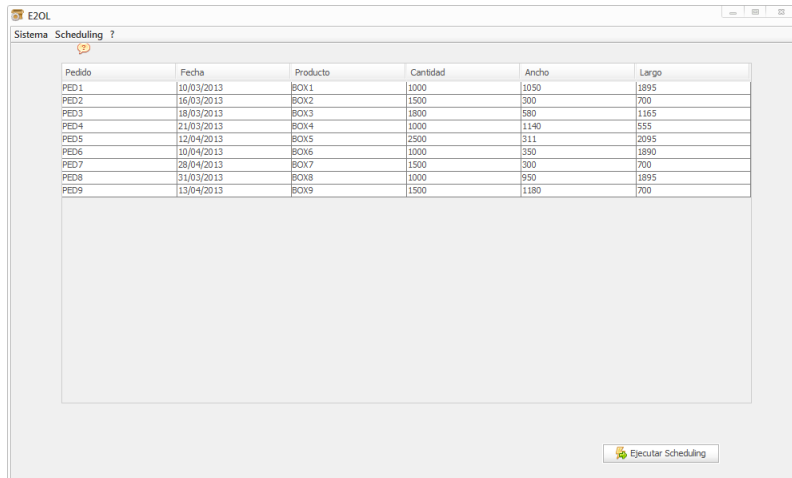
Cuando un encargado de ventas de Todo Cartón efectúa una venta, guarda los datos del cliente, del pedido y del producto en el ERP.

Por otro lado, el encargado de la producción es aquella persona que efectivamente utiliza E2OL para evaluar posibles planificaciones y *schedulings* para la producción, considerando los pedidos en el rango de fechas que solicita. A continuación, se muestran algunas de las interfaces utilizadas en el caso de estudio.

Dado que *Todo Cartón* trabaja con la modalidad *make-to-order*, la Figura 8 detalla la GUI donde se listan las órdenes de ventas, dentro de un rango de fechas determinado.

Al presionar el botón *Ejecutar Scheduling*, el sistema busca los parámetros de configuración que debe cargar el usuario, y muestra la interfaz dinámica que habilita su visualización y carga.

Una vez configurados los valores, el paso siguiente es la ejecución del modelo. Este proceso incluye: la escritura de los archivos con los datos de los parámetros, la búsqueda de datos y escritura del resto de los archivos de entrada para el modelo, la ejecución y solución del modelo matemático, la lectura de los resultados y la posterior traducción en los gráficos (ver Figura 6).



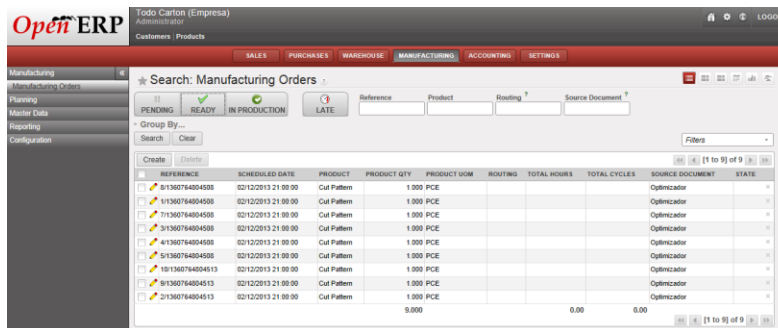
Sistema Scheduling ?

Pedido	Fecha	Producto	Cantidad	Ancho	Largo
PED1	10/03/2013	BOX1	1000	1050	1895
PED2	16/03/2013	BOX2	1500	300	700
PED3	18/03/2013	BOX3	1800	580	1165
PED4	21/03/2013	BOX4	1000	1140	555
PED5	12/04/2013	BOX5	2500	311	2095
PED6	10/04/2013	BOX6	1000	350	1890
PED7	28/04/2013	BOX7	1500	300	700
PED8	31/03/2013	BOX8	1000	950	1895
PED9	13/04/2013	BOX9	1500	1180	700

Ejecutar Scheduling

Figura 8. GUI con la tabla de los datos de las órdenes de ventas.

Finalmente, el usuario decide si guardar o no los resultados del modelo como órdenes de producción en el ERP. En caso positivo, observa un diálogo de confirmación y luego una barra de progreso, mientras este proceso se lleva a cabo. La Figura 9 muestra las órdenes de producción cargadas en *OpenERP*, para el caso de estudio.



OpenERP

Todo Carton (Empresa)
Administrator

Customers Products

SALES PURCHASES WAREHOUSE MANUFACTURING ACCOUNTING SETTINGS

Search: Manufacturing Orders

PENDING READY IN PRODUCTION LATE

Reference Product Routing Source Document

Group By: Search Clear Filters

REFERENCE	SCHEDULED DATE	PRODUCT	PRODUCT QTY	PRODUCT UOM	ROUTING	TOTAL HOURS	TOTAL CYCLES	SOURCE DOCUMENT	STATE
61330764804508	02/12/2013 21:00:00	Cut Pattern	1.000	PCE				Optimizador	...
61330764804508	02/12/2013 21:00:00	Cut Pattern	1.000	PCE				Optimizador	...
71330764804508	02/12/2013 21:00:00	Cut Pattern	1.000	PCE				Optimizador	...
31330764804508	02/12/2013 21:00:00	Cut Pattern	1.000	PCE				Optimizador	...
41330764804508	02/12/2013 21:00:00	Cut Pattern	1.000	PCE				Optimizador	...
51330764804508	02/12/2013 21:00:00	Cut Pattern	1.000	PCE				Optimizador	...
101330764804513	02/12/2013 21:00:00	Cut Pattern	1.000	PCE				Optimizador	...
81330764804513	02/12/2013 21:00:00	Cut Pattern	1.000	PCE				Optimizador	...
21330764804513	02/12/2013 21:00:00	Cut Pattern	1.000	PCE				Optimizador	...
			9.000			0.00	0.00		

(1 to 9) of 9

Figura 9. Órdenes de producción insertadas en el ERP.

5. Conclusiones

E2OL fue desarrollado considerando, en el lado cliente, los módulos pensados por Pinedo (2005), y teniendo en cuenta los problemas de implementación de grandes sistemas que pueden existir en las empresas manufactureras medianas y pequeñas.

La estructura definida permite lograr en buena medida los objetivos propuestos para facilitar el modelado de problemas típicos de producción, integrar los modelos de

optimización matemática con sistemas empresariales (sean tipo ERP o sistemas nativos de desarrollo propio), proveer una manera de incrementar la utilización de dichos modelos en el *planning* y *scheduling* en las empresas manufactureras, y alcanzar el fin de mejorar la gestión en dichas áreas.

A través de la evaluación de E2OL realizada en la ejecución del caso de estudio, se pudo concluir que el sistema utiliza la arquitectura propuesta de forma exitosa, proveyendo una alternativa para ejecutar modelos de optimización de procesos de planeamiento y *scheduling* específicos de la empresa, con la ventaja de ser rápidamente adaptable, personalizable y portable, sin dejar de lado la transparencia para el usuario. Esto a su vez es una característica que distingue a E2OL de otros trabajos, al facilitar una completa integración con los sistemas existentes en la empresa, permitiendo la incorporación y modificación de modelos matemático que consideren las reglas particulares de los negocios, en lugar de proveer modelos genéricos.

Se presentan varias oportunidades de continuar trabajando en el tema, una de ellas es incluir los modelos de optimización de otros sectores de las empresas, tales como carga de camiones, *layouts* de plantas, logística, etc. como consecuencia, también se deberá ampliar la presentación de resultados, incluyendo más opciones de gráficos, o nuevas tablas que para favorezcan el uso de los modelos de optimización.

6. Bibliografía

Bourgeois, S., Artiba, A., Tahon, C. (1993). Integration of Short Term Scheduling with an MRP-II System: Industrial Implementation. Vol. 1, págs. 597-600. ISBN 0-7803-0891-3.

Dirkse, S., Bussiek, M. (2011). *GAMS Data Exchange API*. GAMS Development Corporation.

Framinan, J. M., Ruiz, R. (2010). Architecture of manufacturing scheduling systems: Literature review and an integrated proposal. *European Journal of Operation Research* 205, págs 237-246.

Hvolby, H.-H., Steger-Jensen, K. (2010). Technical and industrial issues of Advanced Planning and Scheduling (APS) systems. *Computers in Industry* 61, págs. 845-851.

Lalas, C., Mourtzis, D., Papakostas, N., Chryssolouris, G. (2005). A combinatorial approach to order release and shop scheduling in discrete manufacturing environments. Págs. 899-906. ISBN 0-7803-9402-X/05.

Laudon, K. C., Laudon, J. P. (2004). *Sistemas de Información Empresariales, Administración de la Empresa Digital*. Pearson Education.

Lee, Y. H., Jeong, C. S., Moon, C. (2002). Advanced planning and scheduling with outsourcing in manufacturing supply chain. *Computers & Industrial Engineering* 43, págs. 351-374.

Masuchun, R., Masuchun, W., & Thepmanee, T. (2009). Integration m-Machine Scheduling into MRP. *4th International Conference on Innovative Computing, Information and Control* (págs. 1365-1368). IEEE Computer Society, 978-0-7695-3873-0/09.

McCarthy, B. L., Liu, J. (1993). Addressing the gap in scheduling research: A review of optimization and heuristic methods in production scheduling. *Internatinal Journal of Production Research* 1, Vol. 31, págs. 59-79.

Montagna, J. M., Leone, H. P. (2000). Proceso de Selección de un ERP en una Empresa Manufacturera Mediana. *First World Conference on Production and Operations Management*.

Muscatello, J. R., Small, M. H., Chen, I. J. (2003). Implementing ERP systems in small and midsize manufacturing firms. *International Journal of Operations & Production Management*, págs. 850-871.

Oracle. (1999). *Design Goals of the Java Programming Language*. Recuperado el 20/01/ 2014, de <http://www.oracle.com/technetwork/java/intro-141325.html>

Pinckaers, F., Gardiner, G., Van Vossel, E. (2011). *Open ERP, a modern approach to integrated business management*. Sexta ed.

Pinedo, M. L. (2005). *Planning and Scheduling in Manufacturing and Services*. Nueva York: Springer Science+Business Media Inc.

Pinedo, M. L. (2012). *Scheduling: Theory, Algorithms, and Systems* (Cuarta ed.). Nueva York: Springer.

Rodríguez, M. A., Vecchietti, A. R. (2013). Integrated Planning And Scheduling With Due Dates In Corrugated Board Boxes Industry. *Industrial and Engineering Chemical Research, Vol. 52*, págs. 847-860.

Van Nieuwenhuysse, I., De Boeck, L., Lambrecht, M., Vandaele, N. (2011). Advanced resource planning as a decision support module for ERP. *Computers in Industri 62*, págs. 1-8.