

Diseño de un simulador en Java para calcular los costos de producción con demanda variable*¹

*[Design of a simulator in Java for calculating
production costs with variable demand]*

CARLOS CARVAJAL², LUIS BENITES³ ROBINSON IBARRA⁴,
ANDRÉS CASTILLO⁵, DARÍO QUIROGA⁶

RECIBO: 20.11.2013 – APROBACIÓN: 04.04.2014

Resumen

En todas las empresas de bienes se debe hacer un plan anual para calcular los costos de producción, en el cual se deben tener en cuenta muchas variables (costos de la materia prima, insumos, mano de obra, maquinaria, mantenimiento y empaques) para cada periodo, ya que la demanda es dinámica. Y se presenta el problema, de que se deben hacer muchas operaciones matemáticas en cada periodo y hacer la sumatoria para el año. Por este motivo se cometen muchos errores, lo que puede afectar seriamente la empresa. Para resolver este problema se desarrollaron modelos y

* **Modelo para la citación de este artículo:**

CARVAJAL, Carlos; BENITES, Luis; IBARRA, Robinson, CASTILLO, Andrés & QUIROGA, Darío (2014). Diseño de un Simulador en Java para calcular los costos de producción con demanda variable. En: Ventana Informática No. 30 (ene-jun). Manizales (Colombia): Facultad de Ciencias e Ingeniería, Universidad de Manizales. p. 109-123. ISSN: 0123-9678

- 1 Artículo de investigación científica y tecnológica proveniente del proyecto *Modelos y algoritmos matemáticos y recursivos aplicados a la industria de producción de bienes*, ejecutado en el periodo 02/2012-03/2013, e inscrito en el Grupo de investigación *en Herramientas para apoyo de procesos educativos*, GIHAPED, de la nombre de la Universidad Cooperativa de Colombia, sede Cali.
- 2 MSc Ingeniería de Sistemas. Docente Investigador, Universidad Cooperativa de Colombia (Santiago de Cali, Valle del Cauca, Colombia) Correo electrónico: carlos.carvajal@campusucc.edu.co
- 3 Estudiante Ingeniería de Sistema, Facultad de Ingeniería, Universidad Cooperativa de Colombia (Santiago de Cali, Valle del Cauca, Colombia). Correo electrónico: luisbehe@hotmail.com
- 4 Estudiante Ingeniería de Sistema, Facultad de Ingeniería, Universidad Cooperativa de Colombia (Santiago de Cali, Valle del Cauca, Colombia). Correo electrónico: ibarraibarraibarra@hotmail.com
- 5 Estudiante Ingeniería de Sistema, Facultad de Ingeniería, Universidad Cooperativa de Colombia (Santiago de Cali, Valle del Cauca, Colombia). Correo electrónico: andref_cas@hotmail.com
- 6 Doctorado Sociedad de la Información y el Conocimiento. Docente investigador, Universidad Cooperativa de Colombia (Santiago de Cali, Valle del Cauca, Colombia). Correo electrónico: dario.quiroga@campusucc.edu.co

algoritmos matemáticos de programación dinámica con el Objetivo de diseñar un simulador en java que pueda predecir los costos de producción a partir de una demanda dada que varía en el tiempo. La metodología empleada fue el Desarrollo de Software Orientado a Objetos y el Proceso Racional Unificado con sus diferentes fases, productos y artefacto usando el Lenguaje de Modelamiento Unificado. Los resultados del proyecto son: Los modelos y algoritmos matemáticos, Los diferentes artefactos diagramas, documentos del RUP. El simulador en Java con sus interfaces, menús, botones, etc.

Palabras Clave: *Demanda, costos de producción, programación dinámica, desarrollo de software orientado a objetos, simulador en java.*

Abstract

All goods companies must make an annual plan to calculate production costs, which must take into account many variables (costs of raw materials, supplies, labor, equipment, maintenance and packaging) for each period, as demand is dynamic. And the problem, they must do many mathematical operations in each period and make the sum for the year is presented. For this reason many mistakes, which can seriously affect the company. To resolve this problem mathematical models and dynamic programming algorithms with the objective of designing a simulator in java that can predict the costs of production from a given demand varies over time developed. The methodology used was the Software Development and Object-Oriented Rational Unified Process with different phases, products and artifact using the Unified Modeling Language. The results of the project are: mathematical models and algorithms, diagrams Different artifacts, documents RU. The simulator in Java with interfaces, menus, buttons etc.

Keywords: *Demand, production costs, dynamic programming, software development, object-oriented, simulator java*

Introducción

Como lo menciona Rivera (2008, 38): según el DANE, en Colombia para el año 2007 (Tabla 1), en el sector industrial había 138 micropymes y 3866 pymes. En el sector metalmecánico había 17 micropymes y 303 pymes, para un total 320, en donde la micropymes son el 4.4 y las pymes el 79.1%.

Tabla 1 - Establecimientos industriales en el 2007 (Adaptada de Rivera, 2008, 38)

Ámbito	Establecimientos	Micros		Pymes		Grandes		Total	
		No	%	No	%	No	%	No	%
Colombia	Industriales	138	1.8	3866	51.4	3520	46.8	7524	100
	Metalmecánica	17	4.4	303	79.1	63	16.4	383	100
Valle del Cauca	Industriales	17	1.6	446	42.4	589	56.0	1052	100
	Metalmecánica	0	0	34	79.1	9	20.9	43	100

En todas estas micropymes y pymes se deben realizar los cálculos de producción, para asignar recursos y planear los gastos con el fin de implementar políticas de administración efectivas (eficientes y eficaces). Para realizar los cálculos y proyectarlos a un futuro, se deben tener en cuenta tres conceptos (la demanda, los inventarios y los costos de producción) que van articulados.

Para calcular los costos de inventarios para varios periodos (Figura 1), se deben tener en cuenta las variables anotadas por Winston (2008, 846), como son: inventario máximo, compra máxima, costo administrativo, costo unitario y costo de almacenamiento. Estos cálculos son un promedio y en muchos contextos como lo formula Vidal (2002, 109), *la demanda* puede ser dinámicas en el tiempo y por lo que hacer los cálculos de *inventarios*, con el concepto del promedio total presenta muchas desviaciones estándar en cada periodo y los costos de inventarios se desajustan y en un periodo dado los datos no son exactos y se puede caer en el error de producir más y tener sobrecostos, o producir menos generando, como lo expone Lay (2009, 14), generando problemas de costos, clientes insatisfechos y afectando las políticas de justo a tiempo.

A partir de la demanda y los inventarios óptimos, se calculan los *costos de producción* para cada periodo y globalmente con sus respectivas variables: materia prima (cantidad, costo unitario, costo total), maquinaria (horas de uso y capacidad), insumos (cantidad, costo unitario, costo total mano de obra (puestos cantidad, costo) y empaques (empaques por cantidad de producto). Es de tener en cuenta que cada una de las variables anteriores, como mínimo se divide en dos, y cada variable puede tener muchos ítems, lo que conlleva, qué, al hacer los cálculos, por cada periodo y globalmente, se deben manejar una cantidad gigantesca de datos y se vuelvan inextricables.

Teniendo en cuenta lo anterior, se *presenta el problema* que los modelos matemáticos para calcular los costos de inventarios y producción con demandas dinámicas son muy complejos ya que cambian en cada periodo y se toman mucho tiempo, motivo por el cual en muchas pymes

no se hacen o se cometen muchos errores, afectando seriamente los presupuestos y las relaciones con los clientes ya que requieren sus pedidos justo a tiempo como lo expone Lay (2009, 16).

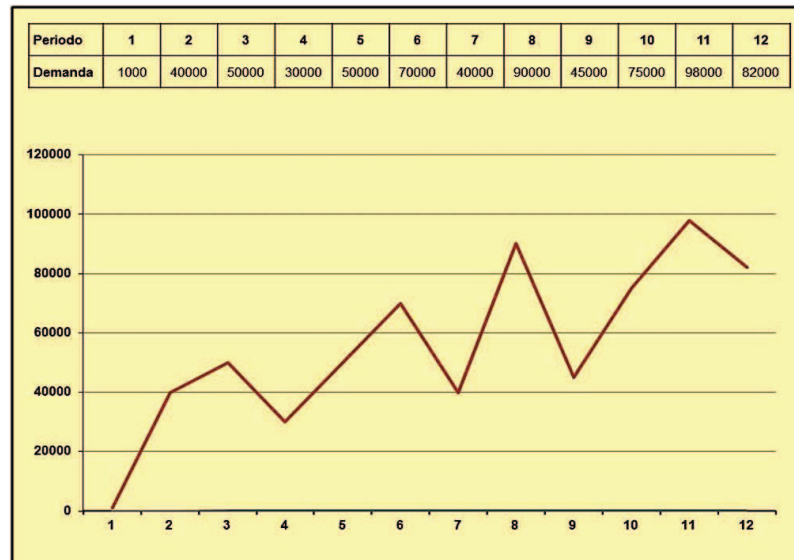


Figura 1. Demanda vs Periodo en una pyme

Para resolver el problema se desarrollaron modelos y algoritmos matemáticos de programación dinámica, *con el Objetivo* de diseñar un simulador en java, que pueda predecir los costos de producción total y por periodo, a partir de un inventario con una demanda dada que varía en el tiempo y sirva para la toma de decisiones en los procesos de las cadenas de producción a nivel administrativo en las pymes.

El diseño del simulador se *Justifica*, ya que, como lo expone Rivera (2008, 38), en Colombia para el 2007 había 138 micropymes y 3866 pymes, en tanto en el departamento del Valle del Cauca existen 446 pymes industriales y 34 metalmecánicas con el 42.4% y el 79.1%. Es importante resaltar que el sector metalmecánico hace un aporte (Tabla 3) de \$5.649.861.810 en el PIB (producto interno bruto), generando 23.479 empleos, representando el 4% del empleos a nacionales. En el ámbito vallecaucano se hace un aporte en el PIB de \$1.259.031.162, generando el 8.2% de empleo. A todas estas empresas se les puede asistir con un software muy económico, ya que en muchos casos, como lo anota Gutiérrez (2009, 125), no lo adquieren porque es muy costoso.

Tabla 3. PIB y Empleo (Adaptada de Rivera, 2008, 38)

Ámbito	Sector	PIB		Empleo	
		Miles \$col	%	Total	%
Colombia	Industriales	108.340.147.335	100	587.630	100
	Metalmecánica	5.649.861.810	5.2	23.479	4
Valle del Cauca	Industriales	15.336.228.830	100	83.256	
	Metalmecánica	1.259.031.162	8.2		

1. Fundamento teórico

Las teorías que se tuvieron en cuenta para el constructo del simulador son: Estadística, Costos de producción, Programación dinámica determinística, Teoría de inventarios determinísticos, Paradigma desarrollo de software orientado a objetos.

1.1 Estadística

Como dice Mendenhall (1997, 2), es la ciencia que se ocupa la colección, clasificación, análisis e interpretación de hechos o datos numéricos. La rama que se dedica a la organización, síntesis y descripción de conjunto de datos es la estadística descriptiva, mientras la estadística inferencial se ocupa de utilizar datos de muestra para inferir algo acerca de una población se denomina.

Suele aplicarse a dos tipos de problemas: - Resumir, describir y explorar datos, y - Utilizar datos de muestra para inferir la naturaleza del conjunto de datos del que se escogió la muestra⁷. En la investigación, la población son las demandas determinísticas del producto (Figura 1) en cada etapa con sus costos y variables.

1.2 Costos de producción

Para Pérez (2010, 51), un proceso es un conjunto de actividades mutuamente relacionadas o que interactúan, las cuales transforman elementos de entrada en resultados o productos. Un producto tiene *«un conjunto de atributos tangibles e intangibles que abarcan empaque, color, precio, calidad y marca, más los servicios y la reputación del vendedor; el producto puede ser un bien, un servicio, un lugar, una persona o una idea»* (Stanton, 2007, 248).

Plantea Chase (2004, 160), que un proceso es una parte de la empresa que toma insumos y los transforma en productos que tendrán un valor más alto que los originales (valor agregado). El propósito de

⁷ Población es el conjunto de datos que se desean describir, en tanto la muestra es un subconjunto de datos seleccionados de una población.

un sistema de producción es determinar las reglas que puede usar la administración para minimizar los costos asociados con mantener la producción y satisfacer la demanda de un cliente. En lo enunciado por Sinisterra (2007, 13), la producción determinística se basa en planificar los costos de producción a partir de una demanda dada.

Los costos incurridos en el proceso de producción es la suma de todas las erogaciones y cargos para convertir la materia prima en producto terminado. El sistema de costeo por órdenes de producción, enunciado por Gómez (2005, 22), también es conocido como costos por órdenes específicas de producción, por lotes de trabajo o por pedidos de los clientes. Es propio de aquellas empresas cuyos costos se pueden identificar con el producto, en cada orden de trabajo en particular, a medida que se van realizando las diferentes operaciones en esa orden específica, es decir, de empresas que producen sus artículos con base en el ensamble de varias partes hasta obtener un producto final, en el cual los diferentes productos pueden identificarse fácilmente por unidades o por lotes individuales, como las industrias tipográficas, la industria de calzado, del mueble, de juguetes, las fundiciones, los astilleros, industria metalmecánica, las sastrerías etc., de productos por demanda de un cliente.

En la investigación se usó el sistema de *costeo por órdenes de producción*, que se basa en datos históricos de demanda (Tabla 3) y (Figura 1), ya que se aplica en gran número de industrias que trabajan en base en órdenes de producción cuya demanda puede variar en el tiempo (por etapas). Como lo describe Sinisterra (2007, 14), los costos de producción se clasifican en *costos directos* de fabricación que son los costos de las materias primas y los costos de la mano de obra; y los *costos indirectos* que son insumos, maquinaria y empaque.

1.3 Programación dinámica determinística

De acuerdo con Azarang (1996, 2), se utiliza para resolver diferentes problemas de optimización en donde las variable cambian en función del tiempo. Se puede decir, según Lew (2006, 5), que determina la solución óptima de un problema en variables, descomponiéndola en etapas, con cada etapa incluyendo un subproblema de una sola variable. La ventaja en el aspecto de los cálculos es que optimizan una sola variable en subproblemas de *n variables*. La principal contribución de la programación dinámica es el principio de *optimalidad*, un marco de referencia para descomponer el problema en *m* etapas. Los cálculos se hacen en forma recurrente e iterativa, ya que un dato necesita del anterior y recursiva cuando una función necesita de otro posterior.

En la investigación se usó con los siguientes criterios: - La empresa debe determinar al principio de cada periodo, cuántas unidades debe producir, - La capacidad es limitada., tanto de producción como de inventario, - Los pedidos de producción se dan en forma discreta, - La producción de cada periodo se debe cumplir a tiempo con el inventario requerido.

1.4 Desarrollo de software orientado a objetos (DSOO)

Como lo expresa Pressman (2010, 235), tiene como fundamentos la reutilización de los componentes de un software. La Unidad básica es la clase, la cual está conformada por los tipos de datos y los métodos. En la DSOO se usan los conceptos siguientes:

- La abstracción: es la capacidad de una clase para representar un objeto de la naturaleza o en la mente del ser humano;
- El encapsulamiento: es la capacidad que tiene el diseño de una clase para introducir, guardar, modificar y acceder los elementos de la clase en forma segura;
- El polimorfismo: es cuando una clase y sus métodos pueden realizar diferentes operaciones; y
- La herencia: cuando una clase hija, puede tomar todos los elementos de una clase, padre como si fueran propios.

En el proyecto se analizaron diferentes modelos matemáticos y algoritmos para diseñar los métodos y clases, así como los otros elementos como son la persistencia de datos y las interfaces. Dado que en el lenguaje de programación Java, la herencia se hace con un solo padre, se optó por un modelado de clases minimalista y en cascada con un diseño arquitectónico a tres capas (interfaz, programa y base de datos). Con el DSOO se usó El Proceso Racional Unificado (RUP), explicado por Larman (2002, 21), que consta de las siguientes fases, productos y artefactos:

- El Modelado del negocio: es todo lo que tiene que ver con el contexto del problema (planteamiento del problema, diseño de los modelos matemáticos, asignación de responsabilidades);
- Los Requisitos: se efectuó el análisis de los requisitos para una la aplicación, como escritura de casos de uso e identificación de requerimientos funcionales y no funcionales;
- El Diseño: habiendo sacado los requisitos, se elaboraron los artefactos en UML: Objetos, Clases, Bases de datos y demás;
- La Implementación: se procedió a traducir los modelos, diagramas y artefactos, al lenguaje de programación Java y al SQL;

- Las Pruebas: se realizaron las pruebas de caja blanca y caja negra, se diseñó el ejecutable en consola y se escribió el manual del usuario.

Para los planos (diagramas) se utilizó el Lenguaje de Modelamiento Unificado (UML), ya que, como lo anota Braude (2005, 256), fue diseñado para un amplio rango de aplicaciones y actividades enfocándose en tres modelos básicos: *el modelo funcional* representado con diagramas de casos de uso en los cuáles se describe la funcionalidad del sistema desde el punto de vista del usuario; *el modelado de objetos*, representado con diagramas de clases, en donde se describe la estructura de un sistema desde el punto de vista de objetos, atributos, asociaciones y operaciones y *el modelado dinámico* representado con: Diagramas de Secuencia, Diagramas de Gráfica de estado y Diagramas de Actividades, en donde se describe el comportamiento interno del sistema. Los Diagramas de Secuencia describen el comportamiento como una secuencia de mensajes intercambiados entre un conjunto de objetos, mientras que los Diagramas de Gráficas de estado describen el comportamiento desde el punto de vista de estados de un objeto individual, y las transiciones posibles entre ellos.

2. Metodología

La ejecución del proyecto se dividió en dos fases, *en la primera fase se planteó el problema* con todas sus variables y restricciones, *posteriormente* se efectuó una búsqueda bibliográfica en donde se investigaron los modelos de inventarios y se diseñaron los modelos matemáticos con sus operaciones, variables y restricciones; mientras en la segunda, se construyeron los modelos matemáticos se desarrollaron los algoritmos recursivos en programación dinámica, que se probaron con modelos típicos de costos de producción.

2.1 Modelo costos de producción

Se realizaron los modelos matemáticos para costeo por órdenes de producción (Tabla 4), que se clasifican en *costos directos e indirectos*, los elementos básicos que se tienen en cuenta para determinar el costo total del producto manufacturado.

Tabla 4. Ecuaciones para el costeo por órdenes

Costo	Descripción	Ecuación
Costos de materia prima	multiplicación de la cantidad por el costo unitario de cada referencia por la demanda de cada etapa	$costo_total_materia_prima = \sum_{r=1}^n costo_total_referencia_r$
	sumatoria del costo total de producción en cada etapa	$costo_total_referencia_r = \sum_{etapa=1}^t (cantidad_r * costo_unitario_r) * produccion_{etapa}$
Costo de mano de obra	multiplicación del número por el salario de cada referencia por la demanda de cada etapa	$costo_total_mano_obra = \sum_{r=1}^n costo_total_referencia_r$
	sumatoria del costo total de Mano de obra en cada etapa	$costo_total_referencia_r = \sum_{etapa=1}^t (numero_puestos_r * salario_etapa_r) * produccion_{etapa}$
Costo de insumos de obra	multiplicación de la cantidad por el costo unitario de cada referencia por la demanda de cada etapa	$costo_total_materia_insumos = \sum_{r=1}^n costo_total_referencia_r$
	sumatoria del costo total de los insumos en cada etapa	$costo_total_referencia_r = \sum_{etapa=1}^t (cantidad_r * costo_unitario_r) * produccion_{etapa}$
Costo de Maquinaria de obra	multiplicación de producción hora por el costo hora de cada referencia por la demanda de cada etapa	$costo_total_maquinaria = \sum_{r=1}^n costo_total_referencia_r$
	sumatoria del costo total de producción en cada etapa	$costo_total_referencia_r = \sum_{etapa=1}^t (produccion_hora_r * costo_hora_r) * produccion_{etapa}$
Costo de empaque de obra	multiplicación de la cantidad por el costo unitario de cada referencia por la demanda de cada etapa	$costo_total_empaque = \sum_{r=1}^n costo_total_referencia_r$
	sumatoria del costo total de los insumos en cada etapa	$costo_total_referencia_r = \sum_{etapa=1}^t (cantidad_r * costo_unitario_r) * produccion_{etapa}$

Costo	Descripción	Ecuación
Costos totales	Costos primos: combinación de los costos directos (materiales directos y mano de obra directa)	$Costos\ primos = Costos\ de\ materiales\ directos + Costos\ de\ mano\ de\ obra\ directa$
	Costos de conversión: suma de mano de obra y los costos indirectos	$Costos\ conversión = Costos\ de\ mano\ de\ obra\ directa + Costos\ indirectos$
	Costos unitarios: resultado de dividir los costos totales por el número de unidades	$Costo\ unitario = Costos\ totales / número\ de\ unidades$
	Costos totales; los costos directos más los costos indirectos o sumatoria de los costos de cada variable	Costos totales = costos directos + costos indirectos $costo_{totales} = \sum_{costo=1}^c costo\ variable$
	Costo de cada variable: sumatoria del costo de producción de cada una de las etapas de la variable	$costo_{variable\ c} = \sum_{etapa=1}^t costo_{total\ variable\ t}$
	Gran costo total: suma del costo de cada variable	$= \sum_{r=1}^n Costo\ materia\ prima$ $+ \sum_{r=1}^n Costo\ total\ mano\ obra$ $+ \sum_{r=1}^n Costo\ insumos$ $+ \sum_{r=1}^n Costo\ maquinaria$ $+ \sum_{r=1}^n Costo\ maquinaria$

Luego se procedió a la construcción del software en Java, Para esto, se usó el DSOO descrito por Deytel (2012, 302), con el *RU-enunciado* por Larman (2002, 235) con sus diferentes fases y artefactos en UML. Como la investigación tiene un altísimo componente matemático, la metodología *RU* usó las metaheurística citadas por Gunther (2010, 33).

2.2 Modelado del negocio

Se planteó el problema en los términos de la primera fase con los objetivos del proyecto de investigación, los recursos y los resultados esperados. El modelado del negocio se orientó hacia los procesos matemáticos que debe hacer el producto de software, que son los calcular los costos de producir un producto a partir de una demanda; la persistencia de datos; la presentación de datos y resultados (interfaces).

Los procesos que debe efectuar el software son: - efectuar los cálculos de los costos de producción a partir de una demanda dada, - hacer los cálculos óptimos de cada etapa (con sus estados), - efectuar la revisión de cada uno de los datos digitados por el usuario e informar al usuario cuando un dato este errado, y no efectuar los cálculos, hasta, cuando, se corrijan los datos, - presentar los cálculos y los costos de producción de cada periodo con sus ítems respectivos (por costo del ítem a producir, compra, etapa, estado, etc.), - permitir la creación de archivos a partir de los datos digitados (gestión de archivos: abrir, crear nuevo, modificar, guardar como y borrar), e imprimir cada interfaz.

2.3 Requisitos

Se identificaron las tareas y se definieron los responsables teniendo en cuenta los recursos disponibles, Se hizo el análisis de los requerimientos funcionales y no funcionales, se diseñaron los casos de uso y se hicieron los prototipos de la interfaces.

Los requerimientos funcionales son:

- Efectuar los cálculos de los inventarios óptimos a producir, teniendo en cuenta una demanda determinística, para cada periodo y el total de todos los periodos.
- Calcular el costo de producción para cada uno de los periodos y el total de todos los periodos, a partir de los inventarios.
- Presentar por separado los cálculos para cada una de las variables del costo de producción como son: materia prima, maquinaria, insumos y mano de obra; por periodo y la sumatoria de los periodos.
- Efectuar la revisión de cada uno de los datos digitados por el usuario e informar al usuario cuando un dato este errado, y no efectuar los cálculos, hasta, cuando, se corrijan los datos.
- Permitir modificar cada una de las variables y volver a efectuar los cálculos.
- Realizar la gestión de archivos: Abrir, crear nuevo, modificar, guardar como y borrar, para cada una de las variables.
- Imprimir

Los requerimientos no funcionales son: El producto de software se debe diseñar con paquetes, sistema operativo, compiladores, lenguajes de programación, herramientas case y demás, de uso público (*Java*, *Linux*, *MySQL*, *Open office*, *Netbeans*, etc.); debe ser multiplataforma (poder correr en Linux y Windows); La computadora: de tener una configuración mínima de Procesador: 3,2 GHz, Memoria RAM 4 Gb, Disco duro de 400 Gb; y por último El software debe tener: Ventanas, Marco, Menús, Botones y eventos y permitir la impresión de todos sus tabulados.

2.4 Diseño

A partir de los requerimientos se diseñaron los casos de uso, el diagrama de clases y los diagramas de secuencia. También se hizo el modelado por componentes. Para el diseño del sistema se usó la Arquitectura Cliente Servidor (ACS) de tres capas planteada por Ramakrishnan (2007, 222), dado que permite, el aislamiento de los programas y los datos, soporte de varias vistas de usuarios y uso de un catálogo para almacenar la descripción de la base de datos.

El objetivo es separar las aplicaciones de usuario y las bases de datos físicas. Las tres capas del diseño están compuestas del *nivel Interno* que describe la estructura de almacenamiento físico de la base de datos. El esquema interno utiliza un modelo de datos físico y describe todos los detalles del almacenamiento de datos y la ruta de acceso a la base de datos; *el nivel conceptual* que describe la estructura de toda la base de datos, para una comunidad de usuarios. El esquema conceptual oculta los detalles de las estructuras de almacenamiento físico y se concentra en describir las entidades, los tipos de datos y las relaciones, las operaciones de los usuarios y las restricciones; y el *nivel de vista o externo*, que incluye las vistas de usuario y describe la parte de la base de datos en la que un grupo de usuarios en particular está interesado y le oculta el resto de la base de datos

2.5 Implementación

Teniendo en cuenta los diagramas anteriores y los modelos matemáticos construidos en la primera fase, se incorporaron las clases especializadas de los modelos matemáticos, se elaboró la base de datos en SQL, se construyeron las interfaces en java y se hizo el modelo de las capas.

2.6 Despliegue y Pruebas

Se realizó lo concerniente a las pruebas de caja blanca y caja negra usando el concepto de las máquinas de estado finito declarados por Hein (2010, 812), de la siguiente manera:

- Componente: Es la parte del software que se va a probar, el Resultado esperado son los resultados que debe arrojar el sistema
- Los estados son las interfaces, resultados matemáticos algunos menús.
- El lenguaje de entrada son los botones – Ítems (y sus escuchas) pulsadas y los datos digitados.
- Las salidas son el resultado de las operaciones, las señales y algunos contenidos de la interfaces.
- El nuevo estado corresponde a la interfaz y/o evento del sistema.

3. Resultados y discusión

3.1 Descripción de resultados

Los resultados obtenidos se pueden resumir en:

- Un producto de software en Java que simula los costos de producción a partir de una demanda dada variable en el tiempo.
- El modelo computacional para calcular los costos directos de fabricación (costos de las materias primas y los costos de la mano de obra); y los costos indirectos (insumos, maquinaria y empaque). Para cada etapa y la sumatoria de todas las etapas.
- La implementación del DSOO y el RUP, para un software con un alto componente matemático.

3.2 Discusión de resultados

Para implementar un software de simulación como herramientas de apoyo a la toma de decisiones es importante tener datos históricos de demanda variable en los periodos dados, los datos se deben ajustar para asegurar la menor desviación estándar y llevarlos a periodos discretos, para articularlos con el sistema por orden de producción. Luego, se desarrollan los modelos matemáticos con programación dinámica con el fin de calcular el óptimo total, por periodo en forma recursiva. Habiendo realizado el modelo matemático se prueba la simulación con los datos obtenidos en la realidad. Posteriormente, se traducen los modelos matemáticos y se construye el producto de software usando el paradigma de DSOO. Una vez concluido el producto de software, se prueba se comparando los datos procesados por el software, con los obtenidos por el modelo matemático y los de la población muestra.

4. Conclusiones

Las pymes producción de bienes son un gran nicho para el uso de simuladores en sus procesos productivos agregando valor a sus procesos y tomas de decisiones ya que pueden ahorrar recursos, ajustando la producción y sus costos a las demandas históricas.

Los sistemas de manufactura por órdenes de producción tienen características de sistemas discretos ya que ocurren en unos tiempos (etapas) determinados, por lo tanto se le pueden aplicar algoritmos de programación dinámica para su planeación.

Los procesos de producción tienen características de fenómenos recursivos ya su simulación y planeación se puede hacer desde la etapa final descomponiendo el costo total en sus variables e ir de la última etapa a la primera y de la primera a la última.

Referencias bibliográficas

- AZARANG, Mohammad R. & GARCÍA DUNNA, Eduardo (1996). Simulación y Análisis de Modelos Estocásticos. México (México): Mc Graw Hill / Interamericana de México. 250 p. ISBN: 9789701011737
- BRAUDE, Eric (2005). Ingeniería de Software: Una Perspectiva Orientada A Objetos. 2 ed. Madrid (España): Alfaomega. 539 p. ISBN 9789701508510.
- CHASE, Richard (2004). Administración de la Producción y Operaciones para una Ventaja Competitiva. 10 ed. México (México): McGraw-Hill. 848 p. ISBN: 9789701044681.
- DEYTEL, Paul (2012). Java How to Program. 9 ed. Upper Saddle River (NJ, USA): Prentice Hall. 1496 p. ISBN: 132575663.
- GÓMEZ, Óscar (2005). Contabilidad de Costos. 5 ed. Santafé de Bogotá (Colombia): Mc Graw Hill. 446 p. ISBN: 958-41-0360-1.
- GUNTHER, Zapfel (2010). Metaheuristic Search. New York (NY, USA): Springer. 324 p. ISBN: 9783642113420.
- GUTIÉRREZ, Valentina (2009). Reseña del software disponible en Colombia para la gestión de inventarios en cadenas de abastecimiento. En: Estudios Gerenciales, Vol, 25, No. 110 (enero-mar). Cali (Colombia): Universidad Icesi. p. 125-153. ISSN: 0123-5923. <http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0123-59232009000100007> [consulta: 23/09/2013]
- HEIN, James (2010). Discrete Structures, Logic, and Computability. 3 ed. Burlington (MA, USA): Jones and Bartlett Publisher. 1009 p. ISBN: 0763772062.
- LARMAN, Craig (2002). UML y Patrones. 2 ed. Madrid (España): Prentice Hall. 590 p. ISBN: 8420534382.
- LAY, Kee. (2009). Just-in-Time Logistics. Farnham (England): Gower. 190 p. ISBN: 978-0-566-08900-8.
- LEW, Art (2006). Dynamic Programming: A Computational Tool. New York (USA): Springer. 379 p. ISBN: 9783540370130.
- MENDENHALL, William (1997). Probabilidad y Estadística para Ingeniería y Ciencias. 4 ed. México (México): Prentice Hall. 1182 p. ISBN: 968-7270-13-6.
- PÉREZ, José (2010). Gestión Por Procesos. 4 ed. Madrid (España): Alfaomega. 336 p. ISBN: 84-7356-198-8.
- PRESSMAN, Roger (2010). Ingeniería de Software. 7 ed. México (México): McGraw Hill. 895 p. ISBN: 84-7829-074-5.
- RAMAKRISHNAN, Raghu (2007). Sistema de Gestión de Bases de Datos. 2 ed. Madrid (España): McGraw Hill. 654 p. ISBN: 9788448156381.

-
- RIVERA, Jorge (2008). Decisiones de financiación de la industria metalmecánica del Valle del Cauca. Estudios Gerenciales, Vol. 24, No. 107 (abr-jun). Cali (Colombia): Universidad Icesi. p. 35–57. ISSN: 0123-5923. <https://www.icesi.edu.co/revistas/index.php/estudios_gerenciales/article/view/259/257> [consulta: 23/03/2014]
- SINISTERRA, Gonzalo (2007). Contabilidad de Costos. Bogotá (Colombia): Ecoe. 424 p. ISBN: 9789586484220.
- STANTON, Willian (2007). Fundamentos de Marketing. 14 ed. (México): Mc Graw Hill. 741 p. ISBN: 9701062019.
- VIDAL, Carlos (2002). Gestión de Inventarios. Cali (Colombia): Universidad del Valle. 432 p. ISBN: 9586708632.
- WINSTON, Waine (2008). Operations Research: Applications & Algorithms. 4 ed. Eagan (MN, USA): Thomson. 1440 p. ISBN: 9788131501900.