



FACULTAD DE INGENIERÍA INFORMÁTICA

TRABAJO DE DIPLOMA

“OPTIVACORTUNI. Sistema informático para resolver problemas de optimización en procesos de corte de materiales”

Autor:

Juan Felipe Medina Mendieta.

Tutores:

Narciso Rubén de León Rodríguez.
Yailen Arencibia Rodríguez del Rey.
Annia Pérez Fernández.

2008

Declaración de autoría

Declaro que soy el único autor de este trabajo y autorizo a la Facultad de Informática en la Universidad de Cienfuegos “Carlos Rafael Rodríguez”, para que haga el uso que estimen pertinente con el trabajo de diploma.

Para que así conste firmamos la presente a los ____ días del mes de ____ del ____.

Nombre completo del autor

Nombre completo del primer tutor

Nombre completo del segundo tutor

Nombre completo del tercer tutor

Los abajo firmantes certificamos que el presente trabajo ha sido revisado según acuerdo de la dirección de nuestro centro y el mismo cumple los requisitos que debe tener un trabajo de esta envergadura referido a la temática señalada.

Firma Tutor

Firma ICT

Firma Vicedecano

*A dos flores, Gretel y Grisel,
A mi madre por haber apostado su vida a un proyecto, sus hijos,
A mi padre por ser el hombre que es,
A mis abuelos por ser ejemplos para mí,
A mi abuela Regla, en mi recuerdo más que nunca,
Y, sobre todo, a mi princesa... Susel.*

Agradecimientos.

Muchas son las personas que han contribuido en esta investigación:

Mildrey, las hermanas Marta y Zenaida, Fefa, Aleida, Margarita y Piedad, por haberme guiado en mis primeros pasos. Los físicos que me enseñaron a pensar, Efrén, Guillermo..., y Filgueira además, por regalarme su sencilla amistad. El matemático Alexis, por sembrar en mí la semillita. Alberto Javier por iniciarme de forma magistral en mis primeras incursiones en la informática.

Oscar por estimularme en el estudio, Mario por su profundidad. Bello, las conversaciones a nivel de pasillo fueron fundamentales en el método Branch and Bound. Aneiros, por abrir mi campo al software libre. Morgan, Albert, Jalbert y Yandy juntos salen buenos frutos, este es también de ustedes. Yailem y Annia por ser excelentes tutoras, y sobre todo por la desinteresada amistad. Profesor Rubén de León, por transmitirme tantos conocimientos, respeto y amistad.

Lidia por el apoyo sentimental, mi padre por las tantas preguntas y Susel por las largas revisiones.

A todas gracias, *muchas gracias*.

Resumen.

La presente investigación, titulada “OPTIVACORTUNI. Sistema informático para resolver problemas de optimización en procesos de corte de materiales”, surge a partir de la necesidad de disponer de un paquete computacional (que integre la modelación matemática del problema, el procesamiento computacional del modelo matemático y la interpretación económica de la solución óptima) para ser utilizado en la optimización del proceso de corte de barras de acero en entidades pertenecientes al Ministerio de la Construcción (MICONS). La misma muestra que en las condiciones tecnológicas actuales de nuestro país es posible desarrollar un programa (con estas características) que tenga influencia positiva en el proceso de corte de barras de acero en estas entidades.

El resultado fundamental es la obtención del sistema informático, “OPTIVACORTUNI”, que reúne en un solo producto: la generación de variantes de corte y de sustitución, la eliminación de variantes de corte dependientes, la modelación matemática del problema, el procesamiento computacional del modelo matemático y la interpretación económica de la solución óptima.

Este, como resultado relevante, muestra un reporte totalmente entendible por los usuarios que permite orientar cómo se debe llevar a cabo el corte de materiales garantizando el desperdicio óptimo.

El interesado, al consultar el trabajo, podrá encontrar información concerniente a aspectos teóricos vinculados al problema del corte de materiales y métodos de solución, y al soporte informático que sirve de plataforma para el desempeño de la investigación.

Índice de contenido

Introducción.....	1
Capítulo 1 - Fundamentación teórica.	7
1.1 Introducción.	7
1.2 Análisis teórico del proceso de corte unidimensional.....	7
1.2.1 El proceso de corte.	7
1.2.2 Descripción del objeto de estudio. Actual flujo del proceso.	9
1.2.3 Alternativas de solución.	10
1.3 Análisis teórico de los métodos de solución.....	10
1.3.1 Ciencias de las matemáticas.	10
1.3.2 Investigación de Operaciones.....	12
1.3.3 Modelación matemática del problema.	13
1.3.4 Programación lineal en enteros.	17
1.3.4.1 Método de Lawer and Bell.....	17
1.3.4.2 Método de Filtros de Balas.....	18
1.3.4.3 Método de los Planos cortantes de Gomory	18
1.3.4.4 Método de Branch and Bound.	18
1.3.4.4.1 Propuesta de implementación.....	19
1.3.4.4.2 Heurísticas desarrolladas.....	20
1.4 Análisis sobre las herramientas informáticas existentes que sirven de soporte para desarrollar el sistema.....	22
1.4.1 Descripción de los sistemas existentes.	23
1.4.1.1 Software vinculados con el tema.....	23
1.4.1.1.1 WinQSB.	23
1.4.1.1.2 QSB.....	24
1.4.1.1.3 STORM.	25
1.4.1.2 Software vinculados con el tema en el MICONS.	25
1.4.1.2.1 VarCort.....	25
1.4.1.2.2 VCORTE.	25
1.4.1.3 <i>Optivacortuni</i> . Solución propuesta.	26
1.4.2 Tendencias, metodologías y/o tecnologías actuales.	27
1.4.2.1 Fundamentos de la metodología utilizada.....	27
1.4.2.1.1 Lenguaje de Modelamiento Unificado (UML).....	27
1.4.2.1.2 Análisis y Diseño Orientado a Objetos de Sistemas Informáticos (ADOOSI-UML) versión 5.0.....	28
1.4.2.1.3 Proceso Unificado de Desarrollo (RUP).....	29
1.4.2.1.4 Metodología a seguir.....	30
1.4.2.2 Fundamentación del lenguaje y software utilizado.....	31
1.4.2.2.1 Paradigma de la programación orientada a objetos.	31
1.4.2.2.2 Lenguajes de programación.....	33
1.4.2.2.2.1 C/C++.....	33
1.4.2.2.2.2 Pascal/Object Pascal	34
1.4.2.2.2.3 Java.....	34
1.4.2.2.3 Herramientas informáticas para el desarrollo de aplicaciones.....	35
1.4.2.2.3.1 Borland Delphi v 7.0.....	35
1.4.2.2.3.2 Borland C++ Builder v 6.0.	35
1.4.2.2.3.3 Borland Java Builder 7.	35
1.5 Conclusiones.	36
Capítulo 2 - Descripción y construcción de la solución propuesta.....	37
2.1 Introducción.	37

2.2	Descripción de la solución propuesta.....	38
2.2.1	Descripción del Modelo del Dominio.....	38
2.2.1.1	Definición de las entidades y los conceptos principales.....	38
2.2.1.2	Representación del modelo del dominio.....	39
2.2.1.3	Reglas del negocio a considerar.....	39
2.2.2	Descripción del sistema propuesto.....	40
2.2.2.1	Concepción general del sistema.....	40
2.2.2.2	Requerimientos funcionales.....	41
2.2.2.3	Requerimientos no funcionales.....	44
2.2.3	Modelo de casos de uso del sistema.....	46
2.2.3.1	Actores del sistema.....	46
2.2.3.2	Casos de uso del sistema.....	47
2.2.3.3	Diagrama de Casos de Uso del sistema.....	49
2.2.3.4	Descripción de los casos de uso del sistema.....	53
2.3	Construcción de la solución propuesta.....	70
2.3.1	Diagrama de clases del diseño.....	71
2.3.2	Diseño de archivos.....	75
2.3.2.1	Diagrama de archivo.....	75
2.3.2.2	Estructura del archivo.....	76
2.3.3	Diagrama de implementación.....	79
2.3.4	Principios de diseño.....	79
2.3.4.1	Estándares en la interfaz de la aplicación propuesta.....	80
2.3.4.2	Tratamiento de errores.....	80
2.3.4.3	Concepción general de la ayuda.....	80
2.4	Conclusiones.....	81
Capítulo 3 - Estudio de factibilidad y validación del sistema.....		82
3.1	Introducción.....	82
3.2	Estudio de factibilidad.....	82
3.2.1	Planificación por puntos de función.....	82
3.2.2	Determinación de los costos.....	89
3.2.3	Beneficios tangibles e intangibles.....	91
3.2.4	Análisis de costos y beneficios.....	92
3.3	Validación.....	92
3.4	Conclusiones.....	95
Conclusiones.....		97
Recomendaciones.....		98
Bibliografía.....		99

Índice de tablas

Tabla 1. -Descripción del actor del sistema-.....	47
Tabla 2. -Caso de uso: Introducir plan de corte para una materia prima en existencia- .	53
Tabla 3. -Caso de uso: Introducir plan de corte para varias materias prima en existencia-	54
Tabla 4. -Caso de uso: Consultar ayuda-	54
Tabla 5. -Caso de uso: Guardar plan de corte-	55
Tabla 6. -Caso de uso: Cargar plan de corte-	56
Tabla 7. -Caso de uso: Calcular variantes de corte para una materia prima en existencia-	56
Tabla 8. -Caso de uso: Calcular variantes de corte para varias materias prima en existencia-	57
Tabla 9. -Caso de uso: Ver reporte referente a las variantes de corte para una materia prima en existencia-	59
Tabla 10. -Caso de uso: Ver reporte referente a las variantes de corte para varias materias prima en existencia-.....	60
Tabla 11. -Caso de uso: Ver reporte de efectividad para una materia prima en existencia-	61
Tabla 12. -Caso de uso: Ver reporte de efectividad para varias materias prima en existencia-	62
Tabla 13. -Caso de uso: Calcular dependencias-.....	63
Tabla 14. -Caso de uso: Suprimir variantes de corte dependientes-.....	63
Tabla 15. -Caso de uso: Reporte de la forma canónica del modelo matemático para una materia prima en existencia-.....	64
Tabla 16. -Caso de uso: Reporte de la forma canónica del modelo matemático para varias materias prima en existencia-	65
Tabla 17. -Caso de uso: Reporte del plan de corte optimizado para una materia prima en existencia-	65
Tabla 18. -Caso de uso: Reporte del plan de corte optimizado para varias materias prima en existencia-	66
Tabla 19. -Caso de uso: Reporte práctico para una materia prima en existencia-	67
Tabla 20. -Caso de uso: Reporte práctico para varias materias prima en existencia-....	67
Tabla 21. -Caso de uso: Optimizar modelo matemático-	68
Tabla 22. -Caso de uso: Llevar modelo matemático de la forma canónica a la estándar-	69
Tabla 23. -Caso de uso: Generar bases de datos referentes a una materia prima en existencia-	69
Tabla 24. -Caso de uso: Generar bases de datos referentes a varias materias prima en existencia-	70
Tabla 25. -Planificación. Entradas externas-.....	83
Tabla 26. -Planificación. Salidas externas-	86
Tabla 27. -Planificación. Peticiones-	87
Tabla 28. -Planificación. Ficheros internos-	88
Tabla 29. -Planificación: Punto de función-	88
Tabla 30. -Planificación: Miles de instrucciones fuentes-	89
Tabla 31. -Costos: Factores de escalas-.....	90

Tabla 32. -Costos totales-91

Índice de Ilustraciones

Ilustración 1. -Flujos de trabajo de la metodología RUP. Requisitos, análisis, diseño, implementación y prueba tienen lugar sobre las cuatro fases-.....	30
Ilustración 2. -Diagrama de clases del modelo de objetos del dominio-.....	39
Ilustración 3. -Diagrama de casos de uso dividido por paquetes-.....	49
Ilustración 4. -Modelo de casos de uso. Paquete opciones-.....	50
Ilustración 5. -Modelo de casos de uso. Paquetes variantes de corte-.....	51
Ilustración 6. -Modelo de casos de uso. Paquetes optimización-.....	52
Ilustración 7. -Diagrama de clases del diseño-.....	71
Ilustración 8. -Diagrama de clases del diseño. Parte 1-.....	72
Ilustración 9. -Diagrama de clases del diseño. Parte 2-.....	73
Ilustración 10. -Diagrama de clases del diseño. Parte 3-.....	74
Ilustración 11. -Diagrama de clases persistentes. Plan de corte-.....	75
Ilustración 12. -Diagrama de clases persistentes. Variantes de corte para una materia prima en existencia-.....	75
Ilustración 13. -Diagrama de clases persistentes. Variantes de corte para varias materias prima en existencia-.....	76
Ilustración 14. -Estructura del archivo. Plan de corte-.....	76
Ilustración 15. -Estructura del archivo. Encabezado de plan de corte-.....	77
Ilustración 16. -Estructura del archivo. Dato de plan de corte-.....	77
Ilustración 17. -Estructura del archivo. Variantes de corte para una materia prima-.....	78
Ilustración 18. -Estructura del archivo. Encabezado de variantes de corte para una materia prima-.....	78
Ilustración 19. -Estructura del archivo. Cantidad de variantes de corte del archivo variantes de corte para una materia prima-.....	78
Ilustración 20. -Estructura del archivo. Datos del plan de corte del archivo variantes de corte para una materia prima-.....	78
Ilustración 21. -Estructura del archivo. Campo VC del archivo variantes de corte para una materia prima-.....	78
Ilustración 22. -Diagrama de implementación-.....	79
Ilustración 23. -Entrada de datos para una situación real-.....	93
Ilustración 24. -Modelación matemático para una situación real-.....	94
Ilustración 25. -Reporte práctico para una situación real-.....	95

Introducción.

El corte unidimensional de materiales se presenta como un proceso **frecuente y común** en la industria.

Frecuente, debido a la sistematicidad con que se realiza. Común, pues se manifiesta en diferentes sectores industriales influyendo en ello de manera significativa la llamada Revolución Industrial, como medio de evolución de una sociedad desde una economía tradicional hasta otra caracterizada por la mecanización^[1].

En el corte de materiales considerando una sola dimensión se encuentran presentes tres entidades fundamentales: **materia prima**, **piezas** que se desean obtener y **variantes de corte** a llevar a cabo.

La materia prima, para el caso que se analiza, se define como cualquier material que necesite ser cortado tomando en cuenta sólo una de sus dimensiones. En general, consiste en materiales procesados industrialmente con tamaños estándares a fin de proveer a la mayor cantidad de clientes (ejemplo: barras de acero, listones de madera, cristales, rollos de tela). Una vez que la materia prima se encuentra en las entidades, esta es cortada en las piezas que realmente son necesarias para el desarrollo del proceso productivo de las mismas.

Existen diferentes combinaciones para realizar el corte de la materia prima. Estas dependen de la longitud de la dimensión a tomar en cuenta para realizar el corte y de la longitud de cada pieza a obtener. Esas diferentes formas o combinaciones posibles se denominan variantes de corte.

Cuba, a consecuencia del desarrollo alcanzado, no se encuentra exenta del corte unidimensional de materiales en la industria: así el Ministerio de la Construcción (MICONS), el Ministerio de la Industria Básica (MINBAS), el Ministerio del Azúcar (MINAZ), el Ministerio de la Agricultura (MINAGRI), cuentan dentro de sus procesos productivos con dicha tarea.

^[1]Revolución Industrial. Tomado de: Enciclopedia Encarta @ 2006, 15 de mayo 2008.

En la etapa actual del desarrollo industrial de nuestra provincia se cuenta con talleres que se dedican a la elaboración de piezas prefabricadas, de acero, tela, cristales y otros materiales^[2].

Algunas entidades llevan a cabo la tarea de cortar esas piezas prefabricadas. La Empresa Forestal comprende el corte de tablas y horcones (madera en general) en los diferentes aserríos. La cristalería abarca el corte de espejos, con las dimensiones particulares que cada cliente solicite. El MICONS recibe de la empresa Antillana de Acero barras de acero de longitudes estándares de 12, 10.5 y 9 metros respectivamente. Estas se cortan en piezas que se utilizan en la construcción de edificaciones, viales y otros.

En el MICONS el proceso de corte de acero se lleva a cabo de forma tradicional. Entiéndase así: el jefe de producción le entrega a los operarios o cortadores el plan de corte, que comprende las longitudes de piezas a cortar y la cantidad a obtener por cada tipo; estos desarrollan la labor, a partir de su experiencia empírica, incurriendo (sin proponérselo) en **desperdicios** significativos.

Una vez más la ciencia en función de resolver problemas de la vida práctica, ofrece respuestas al problema que supone la obtención de sobrantes sin utilidad en el proceso de corte de materiales, encontrando formas de **disminuir estos desperdicios**.

La **Investigación de Operaciones (IO)** como rama de las matemáticas, brinda métodos de solución consistentes en el uso de modelos matemáticos y algoritmos con el fin de realizar un proceso de toma de decisiones, teniendo en cuenta la escasez de recursos para determinar cómo se pueden maximizar o minimizar los mismos^[3].

Mediante la IO se puede modelar matemáticamente la problemática de cortar una materia prima, obteniendo el mínimo desperdicio (o desperdicio óptimo). Esta modelación se conoce como problemas de corte.

Los problemas de corte se basan en la **programación lineal en enteros**, siendo cada variante de corte una variable y los desperdicios que acarrear dichas variantes, coeficientes de la función objetivo a minimizar. Estos problemas pueden llegar a contar con una **gran cantidad de variables** puesto que pueden existir múltiples

^[2]de León Rodríguez, Narciso Rubén. Métodos Matemáticos en la Dirección de los Procesos de Corte de Materiales.--Tesis en Opción al Grado Científico de Master en Matemática Aplicada: UCF (Cf), 1996.-h.3.

^[3]Investigación de Operaciones. Enciclopedia Wikipedia. Tomado de: <http://es.wikipedia.org/IO>, 12 de mayo 2008.

combinaciones de corte (en el orden de cientos y miles en ocasiones) a ser aplicadas en una materia prima.

El jefe de producción del MICONS, aunque cuenta con una herramienta (la utilización de la *modelación y solución* matemática del problema) para **orientar el proceso de corte** de los operarios con el fin de obtener un desperdicio óptimo, se ve imposibilitado de aplicarla, debido a la **gran talla de los problemas de corte** a los que se enfrenta.

En la Universidad de Cienfuegos, durante la década del 90, surgieron trabajos^[4] a partir de la utilización de métodos matemáticos, con el fin de **disminuir la cantidad de variables** en estos problemas, sin que se pierda el sentido matemático de los mismos.

Estos trabajos, aprovechando el desarrollo de los poderosos medios de cálculo electrónicos, devinieron en **sistemas** que calculaban las variantes de corte y reducían de manera significativa las mismas; abarcando sólo estos dos aspectos, debido a que necesitaban un alto uso de los recursos de los computadores (memoria y capacidad de procesamiento), los cuales eran limitados (para esos problemas de tanta magnitud). Luego, con la salida de estos sistemas, se debía formular el modelo matemático, procesarlo e interpretar la solución alcanzada.

Los ordenadores, en esa etapa, contaban con microprocesadores cuyas capacidades de procesamiento eran muy inferiores a las actuales. La memoria, en el orden de los mega bites (MB), se manifestaba insuficiente para almacenar la gran cantidad de variantes de corte que se generaban, por lo que se hacía necesario implementar técnicas de almacenamiento dinámico en memorias externas (discos duros), que a su vez incrementaban la cantidad de procesamiento. Implicaba, sin dudas, un alto nivel de complejidad el hecho de implementar estos problemas, explotando obligatoriamente al máximo los recursos computacionales.

Estos sistemas se pusieron a disposición de los directivos en el MICONS proporcionando valiosas soluciones a problemas concretos. Sin embargo, se conoce que el corte sigue desarrollándose de forma tradicional, debido a limitaciones en la aplicación práctica de los resultados de dichos sistemas por dificultades en:

- **La modelación matemática del problema.**
- **El procesamiento computacional del modelo matemático.**

^[4]de León Rodríguez, Narciso Rubén. Métodos Matemáticos en la Dirección de los Procesos de Corte de Materiales.--Tesis en Opción al Grado Científico de Master en Matemática Aplicada: UCF (Cf), 1996.-h.1.

- **La interpretación económica de la solución óptima del modelo.**

Hoy en día los **avances tecnológicos** experimentados en la rama de la informática, en especial en cuanto a hardware se refiere, donde se duplica y triplica la capacidad y tiempo de cómputo, contándose además con memorias en el orden de los giga bites (GB), de un acceso mucho más rápido, permiten que, ante problemas de gran magnitud, los recursos de los ordenadores no se presenten limitados.

Es por ello que, y aprovechando estos avances, se identifica como **problema a resolver** la necesidad de disponer de un sistema informático (que integre la modelación matemática del problema, el procesamiento computacional del modelo matemático y la interpretación económica de la solución óptima) para ser utilizado en la optimización del proceso de corte de barras de acero en entidades pertenecientes al MICONS.

En consecuencia, es **objeto de estudio** el proceso de corte de barras de acero en el MICONS, y más específicamente, el **campo de acción** se presenta como la disminución de sobrantes (desperdicios) al llevar a cabo dichos cortes.

Se plantea como **idea a defender** que la utilización de un sistema informático (que integre la modelación matemática del problema, el procesamiento computacional del modelo matemático y la interpretación económica de la solución óptima) tiene influencia positiva en el proceso de corte de barras de acero (con incidencia directa en la organización y control) en entidades pertenecientes al MICONS.

Como **objetivo general** se plantea, desarrollar un sistema informático (que integre la modelación matemática del problema, el procesamiento computacional del modelo matemático y la interpretación económica de la solución óptima) para ser utilizado en la optimización del proceso de corte de barras de acero en entidades pertenecientes al MICONS.

Del antes mencionado propósito se desprenden los siguientes **objetivos específicos**:

- Analizar las características y conceptos principales del corte.
- Diseñar el sistema informático.
- Implementar la determinación de las variantes de corte.
- Implementar la solución de problemas de programación en enteros.
- Preparar el recurso informático para presentar el informe de resultados.
- Integrar los elementos informáticos que componen el sistema.

Para cumplir los objetivos trazados se desarrollaron las siguientes **tareas**:

- Resumir aspectos de actualidad relacionados con:

- El proceso de corte de materiales en la industria.
- Herramientas informáticas existentes que puedan servir de soporte para desarrollar el sistema.
- Identificar los requerimientos del sistema.
- Preparar el sistema informático.
- Validar el sistema informático.
- Documentar la investigación.

Los **métodos científicos** utilizados son:

- Del nivel teórico:
 - Análisis, síntesis, inducción y deducción para resumir lo investigado y estudiado sobre el proceso de corte de materiales en la industria y las herramientas informáticas existentes que puedan servir de soporte para desarrollar el sistema.
 - El método de algoritmización de procesos (tránsito del lenguaje matemático al lenguaje computacional) asociando técnicas novedosas.
 - El método de tránsito de lo abstracto a lo concreto para establecer relación entre la interpretación de los resultados y la realidad objetiva existente.
- Del nivel empírico:
 - El método de observación para constatar, en determinados momentos de la investigación, el trabajo que se realiza en la industria y/o empresas para optimizar el proceso de corte de materiales.
 - La entrevista a operarios, tecnólogos y directivos como método para confrontar la experiencia empírica y los conocimientos con respecto al posible perfeccionamiento del proceso de corte de materiales.

El **aporte** de esta investigación está dado en que por primera vez se desarrolla en la provincia un sistema informático, que cuenta con la modelación matemática del problema de corte unidimensional, el procesamiento computacional del modelo matemático y la interpretación económica de la solución óptima del modelo integrados como un todo.

Su **importancia práctica** radica en la obtención de un sistema informático que muestre como resultado relevante un reporte totalmente entendible por los usuarios que permite orientar cómo se debe llevar a cabo el corte de materiales garantizando el desperdicio óptimo.

El presente documento se encuentra estructurado en tres capítulos, conclusiones y recomendaciones, además de la bibliografía utilizada.

Los contenidos por capítulo son:

Capítulo 1: Fundamentación teórica:

Se describen los conceptos asociados al dominio del problema, así como las dificultades que se presentan actualmente en relación con el tema y que dan paso a la presente investigación. Se describen los métodos de solución así como aquel que ha sido aplicado. Se analizan las herramientas informáticas existentes que puedan servir de soporte para el desarrollo del sistema.

Capítulo 2: Descripción y construcción de la solución propuesta:

Se describe la solución propuesta utilizando la metodología del Proceso Unificado de Desarrollo de Software (RUP) para definir las entidades y los conceptos principales del entorno en el que trabajará el sistema propuesto. Se hace uso del Lenguaje Unificado de Construcción de Modelos (UML) que utiliza RUP. Se presenta el diagrama de clases del modelo de objetos del dominio y una descripción detallada de las reglas del negocio a considerar. Se detalla la concepción general del sistema propuesto; se plantean los requisitos funcionales y no funcionales; se define el actor y los casos de uso del sistema, el diagrama de casos de uso y la descripción de cada uno de los casos de uso identificados.

Se tienen en cuenta los principios de diseño para la implementación de la interfaz del sistema propuesto y se presentan: el diagrama de clases del diseño, el diagrama del modelo físico y lógico de datos y el diagrama de implementación.

Capítulo 3: Estudio de factibilidad y validación del sistema:

Se describe lo relacionado con la planificación, costo, beneficios tangibles e intangibles, análisis de costo y beneficios en el desarrollo de la aplicación a desarrollar.

Se valida la solución propuesta.

Capítulo 1 - Fundamentación teórica.

1.1 Introducción.

Este capítulo trata todo lo referente a la fundamentación teórica del tema de la investigación, que consiste en el desarrollo de un sistema informático, que tenga implícito la modelación matemática del problema, el procesamiento computacional del modelo matemático y la interpretación económica de la solución óptima del modelo, para resolver problemas de optimización de procesos de corte de materiales en la industria. Se explican los conceptos asociados al dominio del problema, así como las dificultades que se presentan actualmente en relación con el tema y que dan paso a la presente investigación. Se exponen los métodos de solución sustentados en las ciencias de las matemáticas, que comprenden la investigación operativa como vía, incluyendo técnicas heurísticas para la búsqueda de soluciones. Se mencionan herramientas informáticas existentes que sirven de soporte para encontrar soluciones; así como la propuesta realizada, llegando a conclusiones sobre los aspectos en los que esta se diferencia de las otras y las supera. Se realiza, además, un análisis de las tendencias, tecnologías y metodologías actuales, determinándose cuáles van a ser las utilizadas en el desarrollo del sistema.

1.2 Análisis teórico del proceso de corte unidimensional.

1.2.1 El proceso de corte.

El corte de materiales es un proceso que el hombre lleva a cabo desde que empezó a utilizar los recursos que le brinda la naturaleza para explotarlos en su beneficio. En tiempos antiguos, cuando el hombre construía sus implementos de trabajo, caza o viviendas, necesitaba cortar los diferentes materiales (ramas, piedras, hilos) para transformarlos a su gusto.

El corte no presentaba mayores dificultades, puesto que los materiales eran escasos y las necesidades humanas, en relación con la diversidad de objetos, eran pequeñas.

Muchas veces el material era escogido justamente por sus dimensiones, lo que eliminaba la necesidad de cortarlo.

Es con la llegada de la Revolución Industrial que el corte de materiales toma mayores dimensiones.

La primera Revolución Industrial tuvo lugar en Inglaterra a finales del siglo XVIII. Los primeros cambios se realizaron en los procesos de producción, transformando la concepción de lo que se producía y la manera y el lugar donde se hacía. El trabajo pasa de la fabricación de productos primarios a la de bienes manufacturados. Los productos manufacturados aumentaron de forma espectacular debido a las nuevas técnicas. “El crecimiento de la productividad se produjo por la aplicación sistemática de nuevos conocimientos tecnológicos y gracias a una mayor experiencia productiva”^[1].

Con la Revolución Industrial la producción pasa de pequeños talleres a grandes empresa y fábricas, incorporándose máquinas especializadas destinadas a aumentar la eficiencia productiva. El corte de materiales comienza a aparecer en el diario de estas fábricas.

Se manifiestan fundamentalmente tres formas diferentes de realizar el corte de materiales:

- Corte considerando sólo una dimensión del material a cortar y las piezas a obtener (Corte en una dimensión).
- Corte considerando dos dimensiones del material a cortar y las piezas a obtener (Corte en dos dimensiones).
- Corte considerando tres dimensiones del material a cortar y las piezas a obtener (Corte en tres dimensiones).

El corte en una, dos y hasta tres dimensiones puede aplicarse a cualquier material, es el hombre quien decide qué tipo de corte realizar, según sus necesidades.

En el corte de materiales se presentan términos como:

- **Materia prima** – cualquier material que necesite ser cortado.
- **Piezas a obtener** – unidades requeridas como resultado del corte de la materia prima.

^[1]Revolución Industrial. Tomado de: Enciclopedia Encarta @ 2006, 15 de mayo 2008.

- “**Variante o patrón de corte** – forma de cortar una unidad de materia prima en piezas.”^[2]

Todo proceso de corte lleva consigo **el acarreo de desperdicio** de material o lo que se entiende como sobrantes que no tienen ninguna utilidad práctica. El cortador puede influir para que el desperdicio disminuya; puede ser que no halla, pero por lo general cuando se llevan a cabo cortes de un material del que se quieren obtener diferentes tipos de piezas, se hace imposible evitar los sobrantes sin utilidad.

El corte considerando una sola dimensión lineal (de interés en la presente investigación) es aquel en el que se corta un material, no importa cual sea, teniendo en cuenta sólo una de sus dimensiones. Por ejemplo, el corte de tablas con un ancho determinado, en piezas con un ancho específico, manteniendo el largo y el grosor.

Nuestro país, como consecuencia de su desarrollo industrial y con el objetivo de proveer a la mayor cantidad de clientes, comprende la producción de piezas prefabricadas, tales como: barras de acero, tablas y horcones de madera, cristales y espejos, rollos de tela y otros, que dan origen a la necesidad de hacer cortes en estos materiales para adecuarlos a cada entidad en específico. El MICONS, el MINBAS, el MINAZ, el MINAGRI, son ejemplos que cuentan, dentro de sus procesos productivos, con dicha tarea.

En Cienfuegos, varios organismos llevan a cabo el corte de estas piezas prefabricadas. La Empresa Forestal comprende el corte de tablas y horcones (madera en general) en los diferentes aserríos. La cristalería abarca el corte de espejos, con las dimensiones particulares que cada cliente solicita.

Entidades pertenecientes al MICONS reciben barras de acero de longitudes estándares de 12, 10.5 y 9 metros respectivamente. Estas se cortan en piezas que se utilizan en la construcción de edificaciones, viales y otros. Estas entidades sirvieron de objeto de estudio a la presente investigación.

1.2.2 Descripción del objeto de estudio. Actual flujo del proceso.

Actualmente el proceso de corte de barras de acero en entidades pertenecientes al MICONS, organismos donde se realizó la investigación, se lleva a cabo de forma

^[2]de León Rodríguez, Narciso Rubén. Métodos Matemáticos en la Dirección de los Procesos de Corte de Materiales.--Tesis en Opción al Grado Científico de Master en Matemática Aplicada: UCF (Cf), 1996.-h.10.

tradicional. Entiéndase así: el jefe de producción le entrega a los operarios o cortadores el plan de corte, que comprende las longitudes de piezas a cortar y la cantidad a obtener por cada tipo, estos desarrollan la labor, a partir de su experiencia empírica, incurriendo (sin proponérselo) en **desperdicios** significativos.

1.2.3 Alternativas de solución.

El hecho de que el proceso de corte de materiales acarree desperdicios fundamentó la idea de investigar científicamente técnicas y soluciones para disminuir dichos desperdicios.

La modelación matemática en investigaciones científicas ha sido utilizada por el hombre aumentando considerablemente su campo de acción en las esferas del conocimiento.

La I.O. como rama de las matemáticas, brinda métodos de solución en los que se incluye la programación matemática con el fin de realizar un proceso de toma de decisiones, teniendo en cuenta la escasez de recursos.

“La programación matemática constituye un conjunto de métodos que permite tomar decisiones óptimas”.^[3]

1.3 Análisis teórico de los métodos de solución.

Puesto que el corte de materiales incurre inevitablemente en sobrantes, los cuales se interpretan en desperdicio de materia prima que, al llevarse a cabo a gran escala en la industria, tienen una influencia directa en gastos económicos considerables; se repara en la idea de buscar vías de soluciones (de ser posible la mejor) que influyan de manera significativa en la disminución de estos desperdicios. La ciencia de las matemáticas se presenta como herramienta a aplicar para ello.

1.3.1 Ciencias de las matemáticas.

La ciencia (en latín scientia, de scire: conocer), como el “conjunto de conocimientos obtenidos mediante la observación y el razonamiento, sistemáticamente estructurados

^[3]de León Rodríguez, Narciso Rubén. Métodos Matemáticos en la Dirección de los Procesos de Corte de Materiales.--Tesis en Opción al Grado Científico de Master en Matemática Aplicada: UCF (Cf), 1996.-h.1.

de los que se deducen principios y leyes generales”^[4], se ocupa de la búsqueda de conocimientos que, mediante la tecnología, permiten que se lleven a cabo distintas aplicaciones.

En esta búsqueda de conocimientos las ciencias de las matemáticas han jugado un papel esencial.

La matemática, además de una ciencia, es un arte. Comprende las relaciones espaciales y cuantitativas. “Se trata de relaciones exactas que existen entre cantidades y magnitudes, y de los métodos por los cuales, de acuerdo con estas relaciones, las cantidades buscadas son deducibles a partir de otras cantidades conocidas o presupuestas”.^[5]

Las matemáticas surgieron con el fin de realizar los cálculos en el comercio, para hacer mediciones en la naturaleza y para predecir los acontecimientos astronómicos. “Estas tres necesidades pueden ser relacionadas en cierta forma con la subdivisión amplia de las matemáticas en el estudio de la cantidad, la estructura, el espacio y el cambio”.^[6]

Muchas situaciones en la época moderna son resueltas mediante las ciencias de las matemáticas, existiendo una dependencia, muchas veces desapercibida, pero indudable, de nuestra sociedad a la misma. Pero, ¿cómo es posible traducir la gran gama de problemas de naturaleza abstracta del mundo real, al mundo de leyes, fórmulas y números de las matemáticas? Sencillamente: utilizando **variables**.

Las variables constituyen la clave o secreto para moverse constantemente de un mundo a otro, sirviendo de intérpretes o traductores. Ahí radica la importancia que tiene la correcta y rigurosa definición de las mismas.

Una vez que se ha pasado al lenguaje matemático, es necesario contar con una herramienta que permita ilustrar las propiedades y relaciones que se tienen en el mundo real, disponiendo para ello de los **modelos matemáticos**.

Podemos encontrar disímiles modelos matemáticos (resulta una consecuencia del mismo origen de ellos), los cuales deben reflejar el mundo real, mediante reglas, en el mundo matemático. Estos modelos matemáticos son cambiados o procesados para

^[4]Ciencia. Tomado de: Diccionario Encarta @ 2006, 20 de mayo 2008.

^[5]Matemática. Enciclopedia Wikipedia. Tomado de: <http://es.wikipedia.org/wiki/Matem%C3%A1ticas>, 20 de mayo 2008.

^[6]Matemática. Enciclopedia Wikipedia. Tomado de: <http://es.wikipedia.org/wiki/Matem%C3%A1ticas>, 20 de mayo 2008.

llegar a conclusiones (mediante la **interpretación de los resultados**) en la vida cotidiana. Estas técnicas de constante e interminable estudio en las ciencias de las matemáticas es lo que se conoce como **métodos de solución**.

La matemática actual ha sufrido vuelcos rotundos debido al surgimiento y desarrollo de las computadoras. “Este avance ha dado un gran impulso a ciertas ramas de las matemáticas, como el análisis numérico y las matemáticas finitas, y ha generado nuevas áreas de investigación matemática como el estudio de los algoritmos. Se ha convertido en una poderosa herramienta en campos tan diversos como la teoría de números, las ecuaciones diferenciales y el álgebra abstracta. Además, la computadora ha permitido encontrar la solución a varios problemas matemáticos que no se habían podido resolver anteriormente, como el problema topológico de los cuatro colores⁷ propuestos a mediados del siglo XIX”.^[8]

El empleo de modelos matemáticos asociados a técnicas de computación se hace cada vez mayor. Estas herramientas son utilizadas para encontrar soluciones (las mejores de ser posible) a problemas concretos en la economía. Justamente una rama de las matemáticas tiene la característica de intentar encontrar una mejor solución (conocida como solución óptima) para el problema en consideración: la investigación de operaciones.

1.3.2 Investigación de Operaciones.

La Investigación de Operaciones (IO), como su nombre lo dice, significa hacer investigación sobre las operaciones. Se aplica a problemas que se refieren a la conducción y coordinación de actividades (operaciones) dentro de una organización. Se define como una moderna disciplina científica que se caracteriza por la aplicación de teoría, métodos y técnicas especiales para buscar la solución de problemas de administración, organización y control que se producen en los diversos sistemas que existen en la naturaleza y los creados por el hombre.^[9]

⁷También conocido como problemas de coloración de mapas. El teorema dice que cuatro colores son suficientes para dibujar cualquier mapa, con la condición de que dos países fronterizos deben tener distintos colores. Este teorema fue demostrado en 1976 utilizando una computadora de gran capacidad de cálculo en la Universidad de Illinois (Estados Unidos). Se conocen trabajos realizados por estudiantes de la carrera de informática del curso 2003-2004 de la Universidad Cienfuegos que abordan este tema.

^[8]Matemática. Tomado de: Enciclopedia @ Encarta 2006, 3 de abril 2008.

^[9]Introducción a la investigación de operaciones (Tomo 1).--Ciudad de La Habana: ..[s.n.], ..[200?].—p.3.

La IO remonta sus inicios a principios de la Segunda Guerra Mundial donde se hacía necesario asignar recursos escasos a las distintas operaciones militares y a las actividades dentro de las mismas. Una vez concluida la guerra muchos de los éxitos alcanzados en la IO se comenzaron a extrapolar con fines económicos.

Son varios los factores que influyeron en el rápido desarrollo de esta rama aunque se pueden destacar como relevantes dos.

1. El interés levantado en la comunidad mundial de científicos, debido a los logros alcanzados en operaciones militares, por buscar resultados sustanciales en la IO. Como ejemplo sobresaliente de frutos alcanzados gracias a este interés se puede destacar la aparición del **método simplex** desarrollado por George Dantzig¹⁰.
2. La llegada de la “**revolución de las computadoras**”. Para manejar de una manera efectiva los complejos problemas inherentes a esta disciplina por lo general se requiere un gran número de cálculos. Llevarlos a cabo a mano puede resultar casi imposible.^[11]

Los problemas de corte pueden ser modelados mediante la IO haciendo uso de la programación matemática.

“La programación matemática constituye un conjunto de métodos que permite tomar decisiones óptimas, sobre la base de determinar el óptimo de una función de varias variables, seleccionando entre las soluciones posibles las que se obtienen al considerar las restricciones funcionales correspondientes”.^[12]

La programación lineal, como parte de la programación matemática, puede ser utilizada para la modelación de estos tipos de problemas.

1.3.3 Modelación matemática del problema.

El problema general de la programación lineal consiste en dada una función lineal de n variables, determinar valores no negativos de las mismas que hagan óptimo (mínimo

¹⁰George Bernard Dantzig (8 de noviembre de 1914 – 13 de mayo de 2005) fue un matemático reconocido por desarrollar el método simplex y es considerado como el padre de la programación lineal. Recibió honores, tales como la Medalla Nacional a la Ciencia en 1975 y el premio a la teoría John von Neumann en 1974.

^[11]Introducción a la investigación de operaciones (Tomo 1).--Ciudad de La Habana: ..[s.n.], ..[200?].—p.2.

^[12]de León Rodríguez, Narciso Rubén. Métodos Matemáticos en la Dirección de los Procesos de Corte de Materiales.--Tesis en Opción al Grado Científico de Master en Matemática Aplicada: UCF (Cf), 1996.--h.1.

o máximo) el valor de la función lineal, sujeto a determinadas condiciones que asume la forma de un sistema de ecuaciones y/o inecuaciones lineales.

Los problemas de corte pueden ser modelados como problemas de programación lineal de la siguiente manera: cada patrón de corte se convierte en una variable, la función lineal (conocida también como objetivo) siempre a minimizar se forma con los desperdicios de cada patrón de corte y las restricciones se componen con las demandas por tipo de pieza y las cantidades que aparecen en aquellos patrones de corte que incluyen dichas piezas.^[13]

La presente investigación incorpora una propiedad poco utilizada, que permite disminuir la cantidad de patrones de corte haciendo uso de variantes de sustitución. Estas variantes se forman “de combinar una pieza sustituyéndola por las de longitud menor que ella”.^[14] Además, el sistema de restricciones se representa mediante ecuaciones, uno de los resultados que brinda el uso de las variantes de sustitución.

Por lo tanto el modelo se define de la siguiente manera:^[15]

$$\text{MIN } DMP = \sum_{k=1}^r \sum_{j=1}^n d_{jk} \cdot x_{jk} + \sum_{k=1}^r \sum_{u=n+1}^{n+t+1} b_{uk} \cdot x_{uk}$$

SUJETO A:

$$\sum_{k=1}^r \sum_{j=1}^n P_{ijk} \cdot x_{jk} + \sum_{k=1}^r \sum_{u=n+1}^{n+t+1} a_{iuk} \cdot x_{uk} = N_i \quad ; \quad i = 1; \dots; m$$

$$\sum_{j=1}^{n_k} x_{jk} \leq Q_k \quad ; \quad k = 1; \dots; r$$

$$x_{jk} \geq 0 \quad ; \quad x_{jk} \in \mathbb{Z} \quad ; \quad j = 1; \dots; n$$

$$x_{uk} \geq 0 \quad ; \quad x_{uk} \in \mathbb{Z} \quad ; \quad u = 1; \dots; t$$

^[13]Felipe, Pilar. Programación Matemática I.--Ciudad de La Habana: Empresa Nacional de Producción del Ministerio de Educación Superior, 1983.--p.107.

^[14]de León Rodríguez, Narciso Rubén. Métodos Matemáticos en la Dirección de los Procesos de Corte de Materiales.--Tesis en Opción al Grado Científico de Master en Matemática Aplicada: UCF (Cf), 1996.--h.10.

^[15]de León Rodríguez, Narciso Rubén. Métodos Matemáticos en la Dirección de los Procesos de Corte de Materiales.--Tesis en Opción al Grado Científico de Master en Matemática Aplicada: UCF (Cf), 1996.--h.16.

DONDE:

- n:** Cantidad de variantes de corte. ($n = \sum_{k=1}^r n_k$).
- m:** Cantidad de piezas.
- r:** Cantidad de tipos de longitudes de materia prima.
- Q_k:** Cantidad de unidades de materia prima de longitud tipo k.
- j:** Índice de la variante de corte.
- i:** Índice de la pieza.
- k:** Índice del tipo de longitud de la materia prima.
- N_i:** Necesidad de piezas tipo i.
- DMP:** Desperdicio de materia prima.
- d_{jk}:** Desperdicio de materia prima de longitud tipo k según la variante de corte j.
- P_{ijk}:** Cantidad de piezas tipo i que se obtienen al cortar una unidad de materia prima de longitud tipo k según la variante de corte j.
- x_{jk}:** Cantidad de materia prima de longitud tipo k que es necesario cortar según la variante de corte j.
- Z:** Conjunto de los números enteros.
- u:** Índice de la variante de sustitución.
- t:** Cantidad de variantes de sustitución.
- x_{uk}:** Utilización de la variante de sustitución u en la materia prima de longitud tipo k.
- b_{uk}:** Desperdicio de materia prima de longitud tipo k generado por la variante de sustitución u.
- a_{iuk}:** Coeficiente de utilización de la pieza i en la variante de sustitución u para la materia prima de longitud tipo k.
- n_k:** Cantidad de variantes de corte para la materia prima de longitud tipo k.

Se presenta a continuación por la importancia y vínculo que tiene con la implementación, la metodología para la construcción del modelo.

“METODOLOGÍA PARA LA CONSTRUCCIÓN DEL MODELO:

- Obtener las variantes de corte para cada tipo de materia prima teniendo presente la longitud de cada una de ellas (**L_k**) y el tamaño de las piezas a cortar.

- Obtener las variantes de sustitución de piezas para cada tipo de materia prima. Buscar las variantes de corte tomando hipotéticamente y de forma sucesiva las $m-1$ mayores piezas como materia prima y las restantes más pequeñas que esta, como elementos a cortar. Formar las variantes de sustitución a partir de estas y ubicando el elemento -1 en el lugar correspondiente a la pieza tomada como materia prima.
- Formar la matriz compuesta por las variantes de corte y las variantes de sustitución.
- Determinar las variantes de corte que pueden ser expresadas como la suma de una variante de corte y una variante de sustitución” .^[16]
- Determinar las variantes de sustitución que pueden ser expresadas como la suma de dos variantes de sustitución.
- Plantear la matriz cuyas filas están formadas por las n variantes de corte y las t variantes de sustitución no comprendidas en los dos casos anteriores respectivamente. “Incorporar también una fila con ceros en el lugar correspondiente a las $m-1$ mayores piezas, -1 para la pieza más pequeña y la longitud de esta última como desperdicio. (Esta última incorporación es considerada hipotéticamente como variante de sustitución en la formulación general del modelo).
- Construir el modelo general teniendo en consideración las columnas de las r (tipos de materia prima) matrices obtenidas en el paso anterior” .^[17]

Este modelo se procesa para obtener la solución óptima. Esta solución se define como el “conjunto de patrones de corte que al ser ejecutados satisfacen la necesidad mínima por tipo de pieza teniendo presente la cantidad de materia prima disponible y con el mínimo desperdicio”.^[18] Esta solución, expresada en valores enteros, utiliza métodos de solución de programación lineal en enteros.

^[16]de León Rodríguez, Narciso Rubén. Métodos Matemáticos en la Dirección de los Procesos de Corte de Materiales.--Tesis en Opción al Grado Científico de Master en Matemática Aplicada: UCF (Cf), 1996.--h.17.

^[17]de León Rodríguez, Narciso Rubén. Métodos Matemáticos en la Dirección de los Procesos de Corte de Materiales.--Tesis en Opción al Grado Científico de Master en Matemática Aplicada: UCF (Cf), 1996.--h.18.

^[18]de León Rodríguez, Narciso Rubén. Métodos Matemáticos en la Dirección de los Procesos de Corte de Materiales.--Tesis en Opción al Grado Científico de Master en Matemática Aplicada: UCF (Cf), 1996.-- h.10.

1.3.4 Programación lineal en enteros.

La programación lineal en enteros, también conocida como programación discreta, centra su atención en la solución de problemas donde al menos una de las variables de decisión, en la solución óptima, tome valor entero no negativo.

En el caso que todas las variables deban tomar valores enteros, se está en presencia de un problema de programación en enteros puro. Si al menos una de las variables puede ser continua se nombra mixto.

Los problemas de corte, abordados en la presente investigación, se clasifican en puros, puesto que todas las variables, definidas como la cantidad de materia prima a cortar siguiendo una determinada variante de corte, deben tomar valores enteros (No tendría sentido cortar 34.6 barras de acero).

Los problemas de programación en enteros a simple vista pudieran parecer sencillos. Conociendo además la relativa facilidad con la que están resueltos los problemas continuos (mediante el método simplex), se tiende a pensar que un problema en enteros no es más que un problema continuo donde se aplica un redondeo en la solución final. “Sin embargo, es necesario tener en cuenta que este redondeo puede conducir a soluciones no factibles que distan mucho de la solución óptima entera”.^[19]

En realidad, los problemas de programación en enteros tienen bastantes complicaciones, sobre todo, en la aplicación de los métodos de solución, debido a que no cuentan con un criterio que permita moverse siempre de una solución a otra mejor, como presenta el método simplex.

A continuación se exponen, en breve síntesis, algunos métodos de solución para problemas de programación en enteros, así como la bibliografía donde se puede encontrar mayor información del tema:

1.3.4.1 Método de Lawer and Bell.

“El algoritmo de Lawer and Bell es un método de enumeración parcial, es decir, un método cuyo mecanismo permite la evaluación de sólo un número limitado de soluciones y no todas las posibles, lo que constituirá un proceso muy laborioso para la búsqueda del óptimo”.^[20]

^[19]Felipe, Pilar. Programación Matemática I.--Ciudad de La Habana: Empresa Nacional de Producción del Ministerio de Educación Superior, 1983.--p.283.

^[20]Felipe, Pilar. Programación Matemática I.--Ciudad de La Habana: Empresa Nacional de Producción del Ministerio de Educación Superior, 1983.--p.329.

1.3.4.2 Método de Filtros de Balas.

El método de Filtros de Balas consiste en encontrar soluciones factibles del problema conocido como filtro. Para ello se utiliza el método dual del simplex, asociando las variables duales a las restricciones. Las soluciones factibles del problema filtro, también lo son del problema original. En caso de ser enteras, se analizan si pueden ser posibles óptimas; caso contrario, se ramifican para dar lugar a otras soluciones.^[21]

1.3.4.3 Método de los Planos cortantes de Gomory

El método de los Planos cortantes de Gomory adopta dos formas según se esté en presencia de un problema puro o mixto. Luego, se busca como paso previo la solución del problema continuo utilizando los métodos clásicos del simplex y el dual. Si la solución no es la entera se añaden restricciones conocidas como planos cortantes, formadas por los componentes fraccionarios de los términos independientes. Así se calculan soluciones y añaden planos hasta encontrar la solución óptima entera.^[22]

1.3.4.4 Método de Branch and Bound.

El método de Branch and Bound se escoge en la presente investigación como vía de solución al modelo matemático. Este método busca la solución óptima entera del problema lineal utilizando los métodos simplex (entiéndase simplex y dual) de forma iterativa. Una vez que se encuentra la solución al problema continuo, se añaden restricciones al mismo, las cuales se interpretan como ramificaciones respecto a los valores de las variables que requieren ser enteras, definiéndose dos nuevos problemas. La búsqueda termina cuando se halla una solución entera que no pueda ser superada o cuando se desarrollan todas las soluciones posibles y estas resultan no factibles, en este caso el método no tiene solución.^[23]

Este método se interpreta como un árbol de soluciones, donde cada rama lleva a una posible solución posterior a la actual, con la característica que se encarga de detectar en qué ramificación las soluciones dadas ya no están siendo óptimas, para no seguir desarrollando esa rama y no malgastar recursos y procesos.^[24]

^[21]Felipe, Pilar. Programación Matemática I.--Ciudad de La Habana: Empresa Nacional de Producción del Ministerio de Educación Superior, 1983.--p.283.

^[22]Felipe, Pilar. Programación Matemática I.--Ciudad de La Habana: Empresa Nacional de Producción del Ministerio de Educación Superior, 1983.--p.336.

^[23]Felipe, Pilar. Programación Matemática I.--Ciudad de La Habana: Empresa Nacional de Producción del Ministerio de Educación Superior, 1983.--p.322.

^[24]Branch and Bound. Enciclopedia Wikipedia. Tomado de: http://es.wikipedia.org/wiki/branch_and_bound, 10 de abril 2008.

Dada la informal definición anterior se precisa con mayor exactitud el algoritmo por la importancia que presenta en la investigación (Ver anexo 1).

El algoritmo comienza con una relajación del problema y construye un árbol de soluciones, ramificándose aquellas fraccionarias que sean factibles. Las no factibles son acotadas. Cada solución entera factible es tomada como candidata a ser óptima sustituyendo, si existe, a la anterior. La última solución entera que se almacena es la óptima. La complejidad de este algoritmo es exponencial, aunque en ocasiones, al acotar y ramificar se puede encontrar relativamente rápido las soluciones, el hecho de descomponer puede llevar a problemas de gran tamaño.

Árbol de soluciones: Conjunto de soluciones organizadas jerárquicamente.

Soluciones factibles: Aquellas que dan cumplimiento al problema y mejoran la función objetivo.

Relajación del problema: Solución inicial que no presupone el cumplimiento de las restricciones de valores enteros y otorga un valor siempre mejorable a la función objetivo.

Ramificación: Hecho de convertir un problema en dos nuevos agregando restricciones en cada uno, de modo que si en la solución óptima del inicial la variable x_s que requiere ser entera toma un valor fraccionario, sea: $x_s = n_s + f_s$; $0 < f_s < 1$, $n_s \in \mathbb{Z}$, los dos nuevos problemas se definen como:

I. El problema original más la restricción: $x_s \leq n_s$

II. El problema original más la restricción: $x_s \geq n_s$

Acotamiento: Poda o el no desarrollo de un problema (rama del árbol), debido a que su solución es no factible.

1.3.4.4.1 Propuesta de implementación.

El algoritmo Branch and Bound puede tener éxito en la convergencia si lleva a cabo una buena estrategia de acotamiento. Partiendo de esta base, en la presente investigación se estudiaron los factores que influyen en el acotamiento de una rama:

- El primer factor se refiere al **camino o rama a desarrollar** que puede llevar a una solución óptima. En el trabajo y como aporte, se desarrollaron heurísticas que pueden influir de manera significativa en la decisión del camino tomado.
- El segundo factor se refiere a la **forma de recorrer el árbol**. Es necesario mencionar (sin entrar en ningún tipo de demostración) que si el problema converge, no importa la rama desarrollada ni la forma de recorrido, el valor de la función objetivo es el mismo (el óptimo).

Muchas implementaciones del algoritmo hacen un recorrido a la izquierda o la derecha del árbol, en profundidad. Esto quiere decir que escogen uno de los dos sentidos y empiezan a desarrollar esa rama hasta que encuentran una hoja no factible o sin solución, pasando entonces a la otra rama. Así llegan a recorrer el árbol completo.

Esta vía proporciona muy buenos resultados en el caso que la solución óptima se encuentre en la rama donde se inició la búsqueda, puesto que la otra no hay que explorarla ya que es no factible. Pero, ¿qué sucedería si la solución óptima se encontrara en la rama opuesta? La respuesta es evidente, se llevan a cabo una serie de exploraciones innecesarias obteniéndose árboles de gran tamaño.

La forma de recorrido seleccionada en este trabajo, busca evitar de alguna manera las exploraciones innecesarias partiendo, por supuesto, del hecho de que en ocasiones, dada las características de los problemas, esta vía implicaría mucho más procesamiento que la antes mencionada. Puesto que resulta de gran importancia, para tener como referencia y comenzar el proceso de acotamiento, contar con una solución entera, aunque sea alejada de la solución óptima, se desarrolló la idea de explorar simultáneamente los nodos, surgiendo entonces lo nombrado en esta investigación como **desarrollo del árbol por niveles**.

El desarrollo del árbol por niveles se puso a prueba en problemas reales comparando la cantidad de exploraciones con las realizadas en el recorrido derecha en profundidad. Se obtuvo resultados satisfactorios. Se ha agregado como anexo el algoritmo y la implementación desarrollada, para el lector interesado en el tema (Ver anexo 2 y 3).

1.3.4.4.2 Heurísticas desarrolladas.

“Se denomina heurística a la capacidad de un sistema para realizar de forma inmediata innovaciones positivas para sus fines. La capacidad heurística es un rasgo característico de los humanos, desde cuyo punto de vista puede describirse como el

arte y la ciencia del descubrimiento y de la invención o de resolver problemas mediante la creatividad y el pensamiento lateral o pensamiento divergente”.^[25]

La divulgación del concepto se debe al matemático George Pólya, con su libro *How to solve it (¿Cómo resolverlo?)*. “Cuatro ejemplos extraídos del libro ilustran el concepto mejor que ninguna definición:

- Si no consigues entender un problema, dibuja un esquema.
- Si no encuentras la solución, haz como si ya la tuvieras y mira qué puedes deducir de ella (razonando hacia atrás).
- Si el problema es abstracto, prueba a examinar un ejemplo concreto.
- Intenta abordar primero un problema más general (es la paradoja del inventor: el propósito más ambicioso es el que tiene más posibilidades de éxito).^[26]

Como se refirió en el acápite anterior, se desarrollaron heurísticas (de manera empírica se siguieron los pasos propuestos por George Pólya en su libro *¿Cómo resolverlo?*) con el objetivo de encontrar criterios que determinen la rama a explorar que lleve al camino óptimo en el algoritmo Branch and Bound. Dentro del conjunto de variables no enteras de una solución factible, esta rama se interpreta como una de ellas. Por tanto, el objeto de estudio en el desarrollo de estas heurísticas fue la elección de la variable no entera a ser procesada.

La forma común que se utiliza en las implementaciones para escoger una variable no entera y desarrollarla es la elección sin considerar ningún criterio, o sea, se selecciona la primera, la segunda o cualquiera dentro del conjunto de valores no enteros. Se nombrará esta forma debido a su popularidad, **clásica**.

Partiendo de este punto se estudiaron propiedades que existen entre los problemas de programación lineal y las soluciones factibles no óptimas que comprenden más de un valor no entero, alcanzándose los resultados que se exponen a continuación:

Heurística # 1.

Más alejado de su valor entero: Desarrollar aquella variable cuya parte fraccionaria se encuentre a mayor distancia de su propio valor entero en problemas con función objetivo de mínimo. Esta heurística se fundamenta en la característica

^[25]Heurísticas. Enciclopedia Wikipedia. Tomado de: <http://es.wikipedia.org/wiki/Heur%C3%ADstica>, 15 de enero 2008.

^[26]Heurísticas. Enciclopedia Wikipedia. Tomado de: <http://es.wikipedia.org/wiki/Heur%C3%ADstica>, 15 de enero 2008.

presentada por los problemas de programación lineal de mínimo, que establecen las variables condicionadas, hacia su menor valor. *Por tanto, escoger aquella más alejada, que el método solo se encarga de atraer las otras.*

Heurística # 2.

Más cercano a su valor entero: Esta heurística es el mismo caso que el anterior, pero para problemas de máximo. Se define para evitar ambigüedades: Desarrollar aquella variable cuya parte fraccionaria se encuentre a menor distancia de su propio valor entero (y a mayor de su siguiente valor entero) en problemas con función objetivo de máximo. Esta heurística se fundamenta en la característica presentada por los problemas de programación lineal de máximo, que establecen las variables condicionadas, hacia su mayor valor. *Por tanto, escoger aquella más alejada de su valor entero más 1 (la más cercana a su valor entero), que el método solo se encarga de atraer las otras.*

Heurística # 3.

Contrario al objetivo: Esta heurística es una generalización para cualquier problema, sea de mínimo o máximo. Se define así: Si el problema tiene función objetivo de mínimo, aplicar la heurística *Más alejado de su valor entero*. Si el problema tiene función objetivo de máximo, aplicar la heurística *Más cercano a su valor entero*.

Heurística # 4.

Mayor d: Desarrollar aquella variable con el mayor valor **d**, donde **d** se define como la menor distancia entre la parte fraccionaria y los dos valores enteros que la acotan. *Por tanto, escoger aquella que más lejana esté, de ser entera.*

Para la implementación del algoritmo se escogió la heurística *Contrario al objetivo*.

1.4 Análisis sobre las herramientas informáticas existentes que sirven de soporte para desarrollar el sistema.

El sistema propuesto supuso el estudio de:

- Software existentes antes de *Optivacortuni* que de alguna manera se encontraban vinculados con el tema.
- Metodologías para un desarrollo eficaz.
- Herramientas adecuadas al problema para la implementación.

A continuación se exponen los temas fundamentales y de interés estudiados en la investigación.

1.4.1 Descripción de los sistemas existentes.

Existen software que apoyan el proceso de encontrar solución en problemas de corte unidimensional. Estos sistemas no se especializan específicamente en estos tipos de problemas, pero pueden utilizarse para obtener resultados parciales en los mismos. Unos implementan métodos de solución a la programación lineal en enteros y otros la generación de variantes de corte. A continuación se mencionan los estudiados.

1.4.1.1 Software vinculados con el tema.

1.4.1.1.1 WinQSB.

Sistema interactivo de ayuda a la toma de decisiones que contiene herramientas muy útiles para resolver distintos tipos de problemas en el campo de la investigación operativa. El sistema está formado por distintos módulos, uno para cada tipo de modelo o problema. Entre ellos se destacan los siguientes:²⁷

- *Linear programming (LP) and integer linear programming (ILP)*: Este módulo incluye los programas necesarios para resolver el problema de programación lineal gráficamente o utilizando el algoritmo del simplex; también permite resolver los problemas de programación lineal entera utilizando el procedimiento de Ramificación y Acotación (Branch and Bound).
- *Linear goal programming (GP) and integer linear goal programming (IGP)*: Este módulo resuelve modelos de programación multiobjetivo con restricciones lineales.
- *Quadratic programming (QP) and integer quadratic programming (IQP)*: Este módulo resuelve problemas de programación cuadrática, es decir, problemas con función objetivo cuadrática y restricciones lineales. Utiliza un método

²⁷Esta documentación puede ser encontrada en la ayuda del WinQSB.

simplex adaptado. Los modelos de IQP se resuelven utilizando algoritmos de ramificación y acotación.

- *Network modeling (NET)*: Este módulo incluye programas específicos para resolver el problema del trasbordo, el del transporte, el de asignación, el problema del camino más corto, flujo máximo, árbol generador y problema del agente viajero.
- *Nonlinear programming (NLP)*: Este módulo permite resolver problemas no lineales irrestringidos utilizando métodos de búsqueda lineal, problemas no lineales con restricciones y función objetivo con penalizaciones sobre el incumplimiento de las restricciones.
- *PERT/CPM*: módulo de gestión de proyectos en los que hay que realizar varias actividades con relaciones de precedencia.

WinQSB utiliza los mecanismos típicos de la interfase de Windows, es decir, ventanas, menús desplegables y barras de herramientas. Por lo tanto el manejo del programa es similar a cualquier otro que utilice el entorno Windows.

El módulo Linear programming (LP) and integer linear programming (ILP) de este paquete se utiliza en la solución óptima del modelo matemático de problemas de corte. Es necesario, por tanto, si se hace uso de este paquete:

1. Calcular las variantes de corte.
2. Calcular las variantes de sustitución.
3. Buscar dependencias entre los patrones de corte.
4. Conformar el modelo no considerando las variantes de corte y de sustitución dependientes.
5. Introducir el modelo en el paquete.
6. Interpretar los resultados una vez que se haya procesado el modelo en el paquete.

1.4.1.1.2 QSB.

Este programa consta de una serie de módulos independientes que recogen los tópicos más importantes de Investigación Operativa. Cuenta con los mismos módulos que el WinQSB y otros como: *Módulo de Cadenas de Markov de parámetro discreto (Mkp.exe)*, *Módulo de Sistemas de Colas con solución analítica (Qa.exe)*, *Módulo de Simulación de Sistemas de Colas (Qss.exe)*. El sistema con un ambiente MS-DOS

cuenta con el módulo *Integer linear programming (ILP.exe)* que se utiliza en la solución del modelo matemático en problemas de corte de forma similar a lo explicado en el caso anterior.

1.4.1.1.3 STORM.

Paquete similar a los anteriores con un ambiente MS-DOS que cuenta con un módulo para la solución del modelo matemático en problemas de corte. La forma de operar es similar a la explicada anteriormente.

1.4.1.2 Software vinculados con el tema en el MICONs.

Entidades pertenecientes al MICONs, cuentan con los paquetes anteriormente mencionados. Además, tiene a su disposición dos paquetes relacionados con los patrones de corte:

1.4.1.2.1 VarCort.

Paquete realizado a principios de la década del noventa en la Universidad de Cienfuegos con el objetivo de generar las variantes de corte primarias en problemas de corte unidimensional. Cuenta con un ambiente MS-DOS. En el momento que fue realizado, las computadoras presentaban restricciones de recursos ante problemas de tanta magnitud.

VarCort genera todas las variantes de corte primarias. Es necesario, por tanto, si se hace uso de este paquete para la solución de problemas de corte:

1. Calcular las variantes de sustitución.
2. Buscar dependencias entre los patrones de corte.
3. Conformar el modelo sin considerar las variantes de corte y de sustitución dependientes.
4. Encontrar la solución óptima del modelo matemático.
5. Interpretar los resultados de la solución óptima.

1.4.1.2.2 VCORTE.

Paquete realizado a mediados de la década del noventa en la Universidad de Cienfuegos con el objetivo de generar las variantes de corte, las variantes de sustitución y eliminar aquellas dependientes, en problemas de corte unidimensional. Cuenta con un ambiente MS-DOS. En el momento que fue realizado, las computadoras (del tipo IBM con memoria de 256 Kb) presentaban restricciones de recursos ante problemas de

tanta magnitud. Se hizo necesario implementar técnicas de almacenamiento dinámico a disco duro.

VCORTE genera todas las variantes de corte primarias, las de sustitución y elimina aquellas dependientes. Es necesario, por tanto, si se hace uso de este paquete para la solución de problemas de corte:

1. Conformar el modelo sin considerar las variantes de corte y de sustitución dependientes.
2. Encontrar la solución óptima del modelo matemático.
3. Interpretar los resultados de la solución óptima.

Si el jefe de producción del MICONS utiliza VCORTE y un paquete que resuelva el modelo matemático como WinQSB, obtiene la solución del problema. Debe, por tanto, ocuparse de utilizar la salida de VCORTE y conformar el modelo matemático, introducir el modelo al WinQSB, e interpretar los resultados arrojados por el WinQSB.

El trabajo que fundamenta la existencia del paquete está a disposición del organismo, desde hace más de media década, pero no es utilizado debido a dificultades en la modelación matemática del problema, el procesamiento computacional del modelo matemático y la interpretación económica de la solución óptima del modelo.

Nota: Se hace la observación de que la generación de variantes de sustitución y la búsqueda de dependencias, referidas como tareas a realizar al hacer uso de algunos de los software anteriormente nombrados, no tienen si se desea, que llevarse a cabo. Sin embargo realizarlas simplifica significativamente el modelo matemático a procesar.

1.4.1.3 *Optivacortuni*. Solución propuesta.

El sistema propuesto tiene por nombre “*Optivacortuni*”. El título hace referencia a **optimización de variantes de corte unidimensionales**.

Este software especializado en encontrar solución a problemas de corte unidimensional, integra la generación de variantes de corte primarias y de sustitución, la búsqueda de dependencias entre las variantes de corte, la eliminación de aquellas variantes de corte dependientes, **la construcción del modelo matemático, el procesamiento computacional del modelo matemático** y por último, y como aporte significativo, **la interpretación económica de la solución óptima del modelo**.

Optivacortuni facilita el desarrollo de tareas que pudieran ser de poca aceptación por parte de personas implicadas en la ejecución de las mismas al no tener los necesarios conocimientos matemáticos e informáticos. A continuación se relacionan:

- Preparación del modelo matemático una vez que se tienen las variantes de corte independientes. (Lo hace internamente y lo muestra al usuario)
- Procesamiento del modelo (de un volumen considerable de datos) en un paquete de programas computacionales. (Los datos los maneja internamente)
- Interpretación de los resultados. Emite un reporte práctico expresado en un lenguaje en total correspondencia con el utilizado en el plan de corte a ejecutar por el operario.

1.4.2 Tendencias, metodologías y/o tecnologías actuales.

El mundo actual de intercambio de información, de crecimiento en las áreas de la ciencia y de sistemas informáticos sofisticados demanda un cuidado especial en el proceso de desarrollo de un software. El tiempo, la calidad y el trabajo en equipo juegan un papel importante en este sentido.

Para poder garantizar estos tres factores es necesario seguir las indicaciones de alguna metodología. Se exponen a continuación tecnologías existentes en este sentido, con el fin de seleccionar y utilizar la más conveniente.

Se presentan las características fundamentales de dos de las metodologías existentes y más conocidas para el desarrollo de software:

1. Análisis y Diseño Orientado a Objetos de Sistemas Informáticos en su versión 5.
2. Proceso Unificado de Desarrollo de Software (RUP).

Se justifican las razones por las cuales se seleccionó como metodología a seguir, para guiar el proceso de desarrollo del sistema propuesto, RUP. Además, se describen las características fundamentales de algunas de las tecnologías existentes para el desarrollo de software, justificando la selección realizada para la implementación del sistema propuesto.

1.4.2.1 Fundamentos de la metodología utilizada.

1.4.2.1.1 Lenguaje de Modelamiento Unificado (UML).

El Lenguaje de Modelamiento Unificado (UML - Unified Modeling Language) “se define como un lenguaje que permite especificar, visualizar y construir los artefactos de

los sistemas de software...Es un sistema notacional (que, entre otras cosas, incluye el significado de sus notaciones) destinado a los sistemas de modelado que utilizan conceptos orientados a objetos“.[28]

UML es un lenguaje que permite modelar, construir y documentar los elementos que forman un producto de software que responde a un enfoque orientado a objetos.

Este lenguaje nace en 1994 por iniciativa de Grady Booch y Jim Rumbaugh uniéndoseles luego Ivar Jacobson. Desde entonces, se ha convertido en el estándar internacional para definir organizar y visualizar los elementos que configuran la arquitectura de una aplicación orientada a objetos.

Con este lenguaje se pretende unificar las experiencias acumuladas sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar.

UML no es un lenguaje de programación sino un lenguaje de propósito general para el modelado orientado a objetos y también puede considerarse como un lenguaje de modelación visual que permite una abstracción del sistema y sus componentes.

A partir del surgimiento de UML, muchas de las metodologías existentes hasta ese entonces fueron adaptadas para utilizar este lenguaje, como es el caso de las metodologías de Análisis y Diseño Orientado a Objetos de Sistemas Informáticos en su versión 5.0.[29]

1.4.2.1.2 Análisis y Diseño Orientado a Objetos de Sistemas Informáticos (ADOOSI-UML) versión 5.0.

“La metodología de ADOOSI-UML versión 5.0, fue creada en el año 2000 en el Centro de Estudios de Ingeniería y Sistemas adscrito al ISPJAE, en la Habana, por las doctoras Sofía Álvarez y Anaisa Hernández. Es una metodología para el desarrollo de aplicaciones con tecnología orientada a objetos y utiliza, como ha sido expresado anteriormente, la notación UML.“[30]

El objetivo principal de esta metodología es facilitar la comunicación con los usuarios y entre los miembros del equipo de desarrollo.

[28] Larman, Graig. UML y Patrones: Introducción al análisis y diseño orientado a objetos (Tomo 1) --La Habana: Editorial Félix Varela, 2004.--p.15.

[29] Arencibia Rodríguez del Rey, Yailem. OrtoGramma. Sistema informático para el trabajo de la ortografía en la enseñanza primaria.--Trabajo de Diploma: UCF (Cf), 2007.--h.29.

[30] Arencibia Rodríguez del Rey, Yailem. OrtoGramma. Sistema informático para el trabajo de la ortografía en la enseñanza primaria.--Trabajo de Diploma: UCF (Cf), 2007.--h.31.

A diferencia de la forma de trabajo tradicional, el análisis, el diseño y el desarrollo se realizan de forma concurrente por lo que las etapas no coinciden con las establecidas en otras metodologías orientadas a objetos.

El proceso de desarrollo se realiza de forma iterativa e incremental, lo que significa que el mismo no se lleva a cabo de una sola vez, sino que se realiza por ciclos. “En cada uno de estos ciclos se comienza con la funcionalidad correspondiente al núcleo central de la aplicación y se van agregando funcionalidades en cada ciclo hasta alcanzar la aplicación final. De manera general, el núcleo central de la aplicación se corresponde con la funcionalidad que determina la arquitectura del sistema.”^[31]

La documentación propuesta por la metodología se rige bajo lo establecido en la norma ISO 9000 y la notación utilizada para describir los artefactos en el análisis y el diseño de los productos es, UML con algunas modificaciones.

1.4.2.1.3 Proceso Unificado de Desarrollo (RUP).

El Proceso Unificado se presenta como el resultado de tres décadas de desarrollo de software. Sigue un camino, desde el Proceso Objectory, luego por el Proceso Objectory de Rational hasta el Proceso Unificado de Rational.

El Proceso Unificado de Desarrollo fue creado, y concebido debido a la experiencia alcanzada, por el mismo grupo de expertos que crearon UML: Ivar Jacobson, Grady Booch y James Rumbaugh en el año 1998.

El objetivo de esta metodología es guiar el proceso de software hacia un producto de alta calidad y que cumpla con los requerimientos de los usuarios dentro de una planificación y presupuesto establecidos. Esta metodología utiliza UML como lenguaje de modelado.

Es un proceso dirigido por casos de uso, centrado en la arquitectura de los sistemas y dividido en pequeños miniproyectos del proyecto original que son evaluados en diferentes etapas de la vida del software (iterativo e incremental).

RUP avanza a través de una serie de flujos de trabajo por los cuales pasa todo proceso de desarrollo de software (Ver Ilustración 1). Además, cubre el ciclo de vida de desarrollo de un proyecto y toma en cuenta las mejores prácticas a utilizar en el modelo de desarrollo de software.

^[31] Arencibia Rodríguez del Rey, Yailem. OrtoGramma. Sistema informático para el trabajo de la ortografía en la enseñanza primaria.--Trabajo de Diploma: UCF (Cf), 2007.--h.29.

A continuación se muestran estas prácticas:

- Desarrollo de software en forma iterativa.
- Manejo de requerimientos.
- Utiliza arquitectura basada en componentes.
- Modela el software visualmente
- Verifica la calidad del software.
- Controla los cambios.

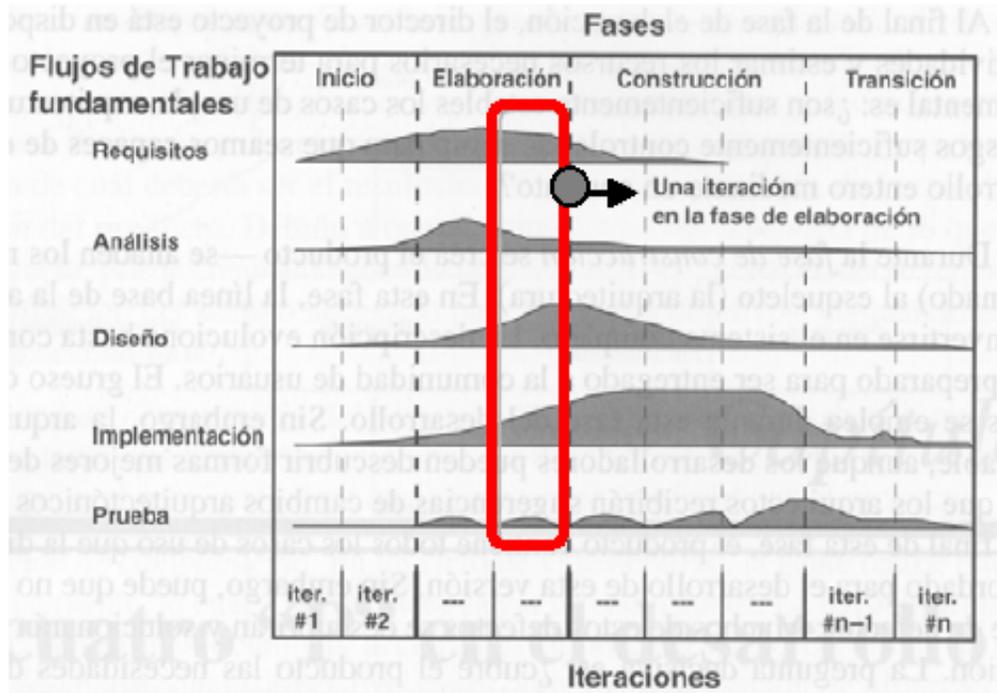


Ilustración 1. -Flujos de trabajo de la metodología RUP. Requisitos, análisis, diseño, implementación y prueba tienen lugar sobre las cuatro fases-

Para apoyar el trabajo con la metodología RUP, ha sido desarrollada por la Compañía norteamericana Rational Corporation la herramienta CASE (Computer Assisted Software Engineering) Rational Rose en el año 2000. Esta herramienta integra todos los elementos que propone el Desarrollo Unificado para cubrir el ciclo de vida de un proyecto.

1.4.2.1.4 Metodología a seguir.

Después de analizar las metodologías: ADOOSI-UML versión 5.0 y RUP, se decidió para la elaboración del presente documento y para llevar a cabo todo el proceso de desarrollo del sistema propuesto, utilizar la metodología RUP.

RUP se encuentra basada en UML y se presenta como el resultado de más de 30 años de experiencia en el desarrollo de sistemas. Se decide seguir esta metodología debido al estándar que representa, reconociendo, además, la calidad y claridad del proceso que propone (bien definido y gestionado), además cuenta con la herramienta CASE Rational Rose (del 2003), que facilita la elaboración de todos los diagramas.

1.4.2.2 Fundamentación del lenguaje y software utilizado.

Para llevar a cabo el desarrollo de la investigación fue necesario el estudio de las tecnologías informáticas y los lenguajes, para seleccionar la más adecuada. La filosofía orientada a objetos fue vital para la reutilización de software que tuvo lugar.

El estudio se realizó bajo el concepto de que el sistema propuesto cuenta con una serie de restricciones y requerimientos. Se mencionan dos de ellos por lo determinante que fueron en la elección del lenguaje de programación para realizar la implementación y de la herramienta de programación utilizada:

1. El sistema realizará una gran cantidad de cálculos y almacenará considerables volúmenes de información en memoria. Por ello resultó importante escoger un lenguaje de programación que permitiera el eficiente uso de los recursos de las computadoras (microprocesador y memoria operativa) en que correrá el sistema.
2. El sistema debe contar con un ambiente de ventanas al estilo Windows. Esto determinó la herramienta de programación a utilizar.

1.4.2.2.1 Paradigma de la programación orientada a objetos.

“La programación orientada a objetos es más que una simple colección de lenguajes de programación nuevos. La programación orientada a objetos es una nueva forma de pensar acerca de lo que significa computar, acerca de cómo podemos estructurar la información dentro de un computador”.^[32]

La programación orientada a objetos es una filosofía que contempla los programas como un conjunto de objetos limitados, con características propias que los describen y obligaciones que determinan su funcionamiento. Es en esencia un acercamiento del

^[32]Budd, Timothy A. Introducción a la Programación Orientada a Objetos.--La Habana: Editorial de Libros Para la Educación, 2005.--p.2.

difícil mundo de la creación de programas, muchas veces abstracto, al mundo *de objetos* que el hombre está acostumbrado a ver y por tanto entiende mejor.

“La programación orientada a objetos (POO) hace modelos de los objetos del mundo real mediante sus contrapartes en software. (...) La programación orientada a objetos nos proporciona una forma más natural e intuitiva de observar el proceso de programación, es decir haciendo modelos de objetos del mundo real, de sus atributos y sus comportamientos. POO también hace modelos de la comunicación entre los objetos. De la misma forma que las personas se envían mensajes uno al otro (por ejemplo, un sargento al comando de tropas para ponerlas en posición firmes), los objetos también se comunican mediante mensajes”.^[33]

Se basa en varias técnicas tales como: herencia, modularidad, polimorfismo, encapsulamiento que permiten expresa un programa como un conjunto de objetos, que colaboran entre ellos para realizar tareas. Esto permite hacer los programas y módulos más **fáciles** de escribir, **mantener** y **reutilizar**.^[34]

Los conceptos principales que supone el paradigma de la programación orientada a objetos son:

- *Clase*: Define las propiedades (características) y el comportamiento (responsabilidades) de un objeto.
- *Objeto*: Entidad provista de un conjunto de propiedades (datos) y de comportamiento (métodos). Se corresponde con los objetos reales del mundo que nos rodea, o a objetos internos del sistema (del programa). Es una instancia a una clase.
- *Método*: Algoritmo asociado a un objeto (o a una clase de objetos), cuya ejecución se desencadena tras la recepción de un "mensaje". Comportamiento que tiene un objeto. Un método puede producir un cambio en las propiedades del objeto, o la generación de un "evento" con un nuevo mensaje para otro objeto del sistema.

Se ha utilizado este paradigma en la investigación aprovechando las ventajas que brinda de: facilidad en la modelación, al verlo todo como objetos, mantenimiento del sistema y reutilización de software.

^[33]Dietel, Harvey M. Como Programar en C-C++ (Tomo 2).--La Habana: Editorial de Libros Para la Educación, ..[199?].--p.595.

^[34]Programación orientada a objetos. Enciclopedia Wikipedia. Tomado de:
http://es.wikipedia.org/wiki/Programaci%C3%B3n_orientada_a_objetos, 10 de mayo 2008.

1.4.2.2.2 Lenguajes de programación.

1.4.2.2.2.1 C/C++

“C es un lenguaje de programación creado en 1972 por Ken Thompson y Dennis M. Ritchie en los Laboratorios Bell como evolución del anterior lenguaje B, a su vez basado en BCPL. Al igual que B, es un lenguaje orientado a la implementación de Sistemas Operativos, concretamente Unix. **C es apreciado por la eficiencia del código que produce** y es el lenguaje de programación más popular para crear software de sistemas, aunque también se utiliza para crear aplicaciones.”^[35]

C es un lenguaje de alto nivel pero con muchas características de medio y bajo nivel. “Dispone de las estructuras típicas de los lenguajes de alto nivel pero, a su vez, dispone de construcciones del lenguaje que permiten un control a muy bajo nivel. Los compiladores suelen ofrecer extensiones al lenguaje que posibilitan mezclar código en ensamblador con código C o acceder directamente a memoria o dispositivos periféricos.”^[36] “C es un lenguaje difícil, que normalmente se enseña solo a programadores experimentados”.^[37]

El lenguaje cuenta con la extensión orientada a objetos C++. “El nombre C++ incluye el operador de incremento (++) de C, para indicar que C++ es una versión mejorada de C.”^[38]

Aunque C/C++ tiene características de bajo nivel es un lenguaje muy popular en la actualidad. Su uso se ha extendido, tanto como lenguaje para el desarrollo de sistemas operativos como de aplicaciones. Muy ligado a UNIX en sus orígenes, hoy día, ligado a otros campos. Las versiones más utilizadas en ordenadores personales son Visual C++ de Microsoft y C++ Builder de Borland (actualmente Inprise Corp.). También ha tenido una gran influencia en el desarrollo del lenguaje JAVA, con el que mantiene similitudes en cuanto a la sintaxis y estructura de la programación.

^[35]Lenguaje de programación C. Enciclopedia Wikipedia. Tomado de: http://es.wikipedia.org/wiki/Lenguaje_C, 27 de enero 2008.

^[36]Lenguaje de programación C. Enciclopedia Wikipedia. Tomado de: http://es.wikipedia.org/wiki/Lenguaje_C, 27 de enero 2008.

^[37]Dietel, Harvey M. Como Programar en C-C++ (Tomo 1) --La Habana: Editorial de Libros Para la Educación, ..[199?].--p.2.

^[38]Dietel, Harvey M. Como Programar en C-C++ (Tomo 2) --La Habana: Editorial de Libros Para la Educación, ..[199?].--p.595.

1.4.2.2.2 Pascal/Object Pascal

Pascal es un lenguaje de programación de alto nivel, diseñado a principios de la década de 1970 por el profesor suizo Niklaus Wirth como un elemento de aprendizaje. Su nombre proviene del matemático francés del siglo XVII, Blaise Pascal, que creó la primera máquina calculadora.

Se encuentra disponible en numerosas distribuciones, tanto interpretadas como compiladas.

Su extensión orientada a objetos Object Pascal, mantiene las ventajas del Pascal, la organización. Por tanto es muy común encontrarlo en ambientes educativos.

La aceptación y el uso de Pascal aumentaron considerablemente en 1984, cuando Borland International introdujo Turbo Pascal. Se trataba de un compilador de Pascal, de alta velocidad y bajo costo, para sistemas MS-DOS.

En los últimos años su popularidad ha decrecido, aunque sigue comercializándose, y se utiliza muy a menudo en medios educativos. El producto comercial actual más conocido es Borland Delphi, un entorno de programación visual, basado en Turbo Pascal para Windows y que usa las extensiones de Object Pascal. Existen también versiones para desarrollos bajo Linux.^[39]

1.4.2.2.3 Java

Java es lenguaje de programación orientado a objetos desarrollado por la empresa Sun Microsystems en 1995. Es un lenguaje de alto nivel muy parecido a C++, pero facilita el trabajo al eliminar los tipos de dato punteros, no visto con muy buenos ojos por algunos programadores. Además presenta marcadas características de seguridad y transportabilidad.

Lenguaje interpretado, que es muy popular en la actualidad debido a su utilización en la programación Web.

“Java incorpora nuevas tareas como un recolector de elementos (...) y elimina aspectos de C++ confusos y muy poco utilizados como la sobrecarga de operadores”^[40]

“Este lenguaje define una máquina virtual independiente de la plataforma donde se ejecuta, que procesa programas, llamados *Applets*, descargados desde el servidor

^[39]Pascal (informática). Tomado de: Enciclopedia Encarta @ 2006, 3 de mayo de 2008.

^[40]Zukowski, John. Programación Java 2 J2SE 1.4 (Tomo 1) .--La Habana: Editorial Félix Varela, 2006.-- p.37.

Web. Además, debido al modo de ejecución de los *Applets*, este lenguaje es muy seguro frente a la presencia y ataque de virus informáticos.”^[41]

1.4.2.2.3 Herramientas informáticas para el desarrollo de aplicaciones.

1.4.2.2.3.1 Borland Delphi v 7.0.

El Object Oriented Pascal es el lenguaje que Delphi utiliza para crear las aplicaciones orientadas a objetos. Borland Delphi es un ambiente de desarrollo rápido de aplicaciones (RAD) muy flexible y fácil de usar. Proporciona una jerarquía muy extensa de clases de objetos reusables. Delphi es una herramienta de propósito general, se puede programar tanto a bajo nivel, como a alto nivel (simplemente usando controles y ajustando propiedades). Las aplicaciones creadas en Delphi sólo funcionan sobre la plataforma de trabajo Windows.

1.4.2.2.3.2 Borland C++ Builder v 6.0.

El lenguaje C++ es el lenguaje que Borland C++ Builder utiliza para crear las aplicaciones. Presenta un ambiente visual de desarrollo para aplicaciones controlado por eventos de usuario sobre interfaces gráficas. Las aplicaciones creadas con esta herramienta sólo funcionan sobre la plataforma de trabajo Windows. El Borland C++ Builder, es un ambiente de desarrollo rápido de aplicaciones (RAD) muy flexible. Dispone de un entorno visual fácil. Permite la importación de código C++ existente. Posee una gran cantidad de clases y objetos reusables. Es una herramienta de propósito general con una gran documentación disponible. ^[42]

1.4.2.2.3.3 Borland Java Builder 7.

Como el Object Pascal y el C++, el Java es un lenguaje orientado a objetos. Las aplicaciones desarrolladas en Java pueden ejecutarse en diferentes plataformas como: Windows, Unix, Macintosh. Incorpora un conjunto de clases que pueden ser usadas en aplicaciones en red, facilitando la creación de aplicaciones distribuidas.

Borland JBuilder 7 ofrece gran flexibilidad para crear aplicaciones Java. Permite el desarrollo de aplicaciones Java en Windows, Solaris o Linux. Incluye un entorno de desarrollo en equipo en su versión Enterprise.^[43]

^[41]Java (informática). Tomado de: Enciclopedia Encarta @ 2006, 3 de mayo de 2008.

^[42]Arencibia Rodríguez del Rey, Yaillem. OrtoGramma. Sistema informático para el trabajo de la ortografía en la enseñanza primaria.--Trabajo de Diploma: UCF (Cf), 2007.--h.36.

^[43]Arencibia Rodríguez del Rey, Yaillem. OrtoGramma. Sistema informático para el trabajo de la ortografía en la enseñanza primaria.--Trabajo de Diploma: UCF (Cf), 2007.--h.37.

1.5 Conclusiones.

La utilización de un sistema informático que integre la modelación matemática del problema, el procesamiento computacional del modelo matemático y la interpretación económica de la solución óptima del mismo se presenta como una alternativa viable en el proceso de corte de materiales en la industria.

La metodología RUP como guía a utilizar en el proceso de desarrollo (por ser estándar), la utilización del lenguaje de programación C/C++ para la implementación (por su eficiencia en el código) y la herramienta informática Borland C++ Builder 6.0 (de desarrollo rápido de aplicaciones) son elementos con influencia positiva en el análisis, diseño y preparación del software a desarrollar en este trabajo de diploma.

Capítulo 2 - Descripción y construcción de la solución propuesta.

Sistema informático *OPTIVACORTUNI*.

2.1 Introducción.

En este capítulo se aborda todo lo referente al **proceso de desarrollo** del sistema propuesto. Proceso que inicia desde su concepción como investigación, para lo cual se hace necesario el profundo conocimiento de las propiedades del dominio que dieron origen al problema científico, hasta su diseño.

Se utiliza la metodología brindada por el Proceso Unificado de Desarrollo de Software (RUP), la cual se encuentra basada en el Lenguaje Unificado de Modelación (UML) y se presenta como el resultado de más de 30 años de experiencia en el desarrollo de sistemas. Se decide seguir esta metodología debido al estándar que representa, reconociendo, además, la calidad y claridad del proceso que propone (bien definido y gestionado).

Con el fin de conseguir un buen entendimiento del funcionamiento del entorno que origina la investigación y utilizando RUP como herramienta:

- se definen las entidades y los conceptos principales del entorno en el que trabajará el sistema propuesto,
- se presenta el diagrama de clases del modelo de objetos del dominio y una descripción detallada de las reglas del negocio a considerar,
- se describe la idea general del sistema propuesto,
- se explica el proceso de captura de los requisitos funcionales y no funcionales necesarios para dar solución al problema planteado,
- se describe el modelo del sistema, a través de la definición de los actores y los casos de uso del sistema, el diagrama de casos de uso y la descripción de cada uno de los casos de uso identificados.

Se expone la forma en que se decide, como resultado de analizar la solución propuesta, llevar a cabo el sistema (diseño). Se plasman los resultados de la etapa de diseño del sistema. Se describe la construcción de la solución propuesta presentando:

- el diagrama de clases del diseño,
- el diseño de archivos desde el punto de vista lógico y físico,
- el diagrama de implementación y los principios de diseño utilizados mostrando los estándares en la interfaz de la aplicación,
- el tratamiento de errores,
- la concepción general de la ayuda.

2.2 Descripción de la solución propuesta.

2.2.1 Descripción del Modelo del Dominio.

El modelo del dominio permite la identificación de los objetos de mayor relevancia en el contexto del sistema (entorno que da origen a la investigación). Según Jacobson “representan las cosas que existen o los eventos que suceden”.^[1] Estos objetos, clases u entidades se identifican en tres formas fundamentales: representando elementos que se manipulan en el negocio, formando parte de conceptos a estudiar por el sistema o como acontecimientos que se darán en el futuro o ya se han dado. La principal forma de describirse consiste en diagramas, los cuales se deben mostrar unificando a clientes y entidades del dominio, junto con sus relaciones expresadas mediante asociaciones.

2.2.1.1 Definición de las entidades y los conceptos principales.

Los objetos y conceptos fundamentales identificados en el entorno que sirvió de fuente a la investigación son:

El **plan de corte** de barras de acero estimado por el jefe de producción del MICONS dada una solicitud, tarea o proyecto. El **cortador u operario** que procede a la ejecución del plan de corte que **le entrega** el jefe de producción. La cantidad de **materia prima** (barras de acero) que debe **ser cortada** por el operario, y la cantidad de cada tipo de **pieza** necesaria que se debe **obtener** según el plan de corte.

^[1]Jacobson, Ivar. El Proceso Unificado de Desarrollo de Software (Tomo 1).--La Habana: Editorial Félix Varela, 2004.--p.112.

2.2.1.2 Representación del modelo del dominio.

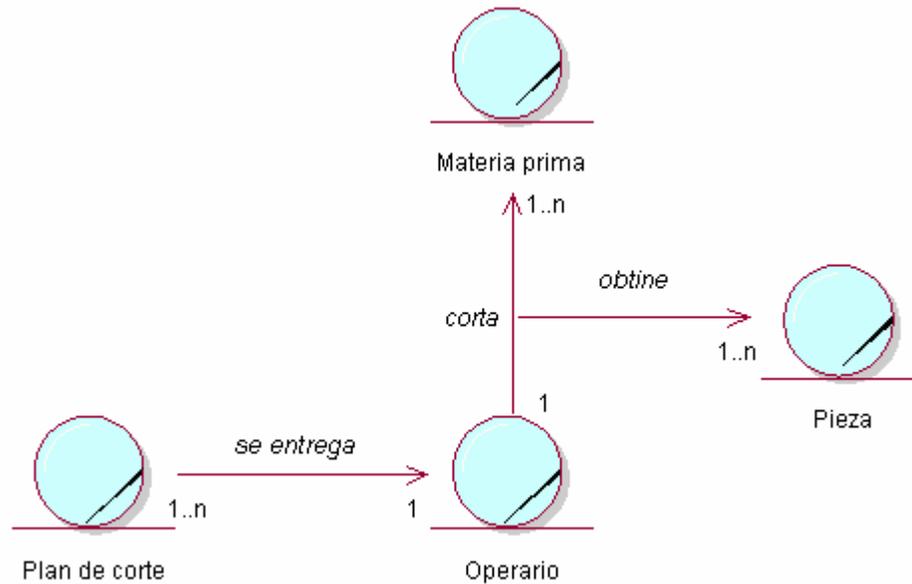


Ilustración 2. -Diagrama de clases del modelo de objetos del dominio-

2.2.1.3 Reglas del negocio a considerar.

Son aquellas que regulan y describen las políticas que deben cumplirse o condiciones que deben satisfacerse para el adecuado funcionamiento del negocio. En el entorno en el que trabajará el sistema se identificaron las siguientes condiciones:

- Los cortes se llevan a cabo a partir de un plan de corte.
- Para llevar a cabo los cortes se toma en cuenta sólo una de las dimensiones de las barras de acero.
- Se verifica que la existencia de barras de acero es suficiente para cumplir con la solicitud de cada tipo de pieza. En caso de no ser así se comunica al jefe de producción para la modificación del plan de corte.
- La longitud de las barras de acero a cortar debe superar la longitud de la pieza mayor requerida.
- El corte de barras de acero se realiza de acuerdo a las longitudes de piezas requeridas.
- Los sobrantes no deben exceder en longitud a la menor de las piezas requeridas.

2.2.2 Descripción del sistema propuesto.

2.2.2.1 Concepción general del sistema.

El resultado a alcanzar con esta investigación es la obtención de un sistema que optimice el proceso de corte de barras de acero en el MINCONS. El sistema propuesto debe presentar un ambiente sencillo, moderno, actualizado; debe ser compatible con los sistemas operativos más populares en nuestro país y específicamente en el MINCONS; con buena estética, no muchos requerimientos de sistema y algoritmos optimizados para que pueda ser utilizado por cualquier usuario, en especial por el jefe de producción del MICONNS, con el objetivo de orientar un proceso de corte que lleve al desperdicio óptimo de barras de acero.

El sistema debe contemplar la opción de aceptar planes de corte que consideren la existencia de un solo tipo de longitud de barra de acero y diferentes tipos de longitudes de barras de acero, de manera que se puedan comparar (si se estima conveniente) los resultados arrojados para uno y para otros, con el objetivo de tomar decisiones que puedan influir significativamente en el proceso de corte.

El sistema debe calcular todas las combinaciones de corte para el caso de una y varias barras de acero en existencia y mostrar una serie de reportes de interés vinculados a las mismas, entre los que se citan: número total de variantes de corte, número total de variantes de corte independientes, número total de variantes de corte dependientes y reportes de efectividad en cuanto a la reducción de patrones de corte.

El sistema debe permitir la **visualización de los modelos matemáticos** obtenidos (tanto para una como para varias barras de acero en existencia) debido a la importancia que esto tiene para la comprensión e interpretación de la solución óptima.

El sistema debe **calcular la solución óptima** para el caso de una y varias barras de acero en existencia y mostrar un reporte con los resultados para pasar a ser interpretado por el cliente del sistema. Este reporte debe permitir verificar, además, si de hecho se cumple o no, el plan de corte requerido.

El sistema debe contar con un **reporte práctico que interprete los resultados**. En el caso de una longitud de barra de acero en existencia, el reporte debe especificar la cantidad de barras a utilizar y los cortes a realizar para cumplir el plan de corte garantizando el desperdicio óptimo. En el caso que se estén considerando varias longitudes de barras de acero en existencia, el reporte debe contar con las indicaciones

de las cantidades de barras por tipo de longitud a cortar y los cortes a realizar para cumplir el plan garantizando el desperdicio óptimo.

El sistema debe permitir guardar y abrir los planes de corte con el objetivo de utilizarlos en momentos posteriores si se desea. Debe contar, además, con una ayuda bien detallada que explique el funcionamiento del mismo.

2.2.2.2 Requerimientos funcionales.

La captura de los requerimientos funcionales implica el estudio de los elementos con los que debe contar el sistema. “Proceso de averiguar, normalmente en circunstancias difíciles, lo que se debe construir”.^[2] Si se establece una escala de uno a diez ascendientemente en dificultad y se califican según la misma, los procesos: captura de los requisitos funcionales y escritura del código, se puede evaluar de la mitad en adelante el primero y no llega a la mitad, el segundo. Paradójicamente el segundo, muy fácil, debe esperar por que esté completado el primero, difícil (los requerimientos, se supone es lo que debe implementar el código). Esta dificultad viene dada puesto que los usuarios, por lo general, no tienen idea de cómo transformar sus trabajos en software.

Es necesario entonces, guiar el trabajo del desarrollo partiendo de tomar en cuenta al cliente (muchas veces con pocos conocimientos informáticos) durante el proceso, hasta el punto en que pueda ser capaz de entender, comprender y leer el resultado de la captura de requisitos. Para ello es necesario utilizar el lenguaje del cliente.

Al utilizar el lenguaje del cliente, los requisitos funcionales permiten determinar, de una manera clara, lo que debe hacer el sistema. Estos tienen como objetivo expresar una especificación detallada de las responsabilidades del sistema que se propone.

Los requerimientos funcionales capturados para el desarrollo del sistema propuesto son los siguientes:

- R1.** Introducir plan de corte para una barra de acero en existencia.
- R2.** Introducir plan de corte para varias barras de acero en existencia.
- R3.** Calcular variantes de corte primarias para una barra de acero en existencia.
- R4.** Calcular variantes de corte de sustitución para una barra de acero en existencia.

^[2]Jacobson, Ivar. El Proceso Unificado de Desarrollo de Software (Tomo 1).--La Habana: Editorial Félix Varela, 2004.--p.105.

- R5.** Calcular variantes de corte primarias para varias barras de acero en existencia.
- R6.** Calcular variantes de corte de sustitución para varias barras de acero en existencia.
- R7.** Calcular dependencias entre las variantes de corte primarias.
- R8.** Calcular dependencias entre las variantes de corte de sustitución.
- R9.** Suprimir las variantes de corte dependientes (aquellas que resultan de la combinación lineal de otras).
- R10.** Elaborar la forma canónica del modelo matemático.
- R11.** Llevar el modelo matemático de la forma canónica a la estándar.
- R12.** Optimizar la forma estándar del modelo matemático.
- R13.** Generar archivo de plan de corte.
- R14.** Cargar archivo de plan de corte.
- R15.** Disponer de una ayuda bien detallada sobre las principales opciones del sistema.
- R16.** Emitir reporte de todas las variantes de corte generadas (primarias y de sustitución) para una barra de acero en existencia.
- R17.** Emitir reporte de todas las variantes de corte primarias generadas para una barra de acero en existencia.
- R18.** Emitir reporte de todas las variantes de corte de sustitución generadas para una barra de acero en existencia.
- R19.** Emitir reporte de todas las variantes de corte independientes generadas (primarias y de sustitución) para una barra de acero en existencia.
- R20.** Emitir reporte de todas las variantes de corte independientes primarias generadas, para una barra de acero en existencia.
- R21.** Emitir reporte de todas las variantes de corte independientes y de sustitución generadas, para una barra de acero en existencia.
- R22.** Emitir reporte de todas las variantes de corte dependientes generadas (primarias y de sustitución) para una barra de acero en existencia.
- R23.** Emitir reporte de todas las variantes de corte dependientes primarias generadas, para una barra de acero en existencia.
- R24.** Emitir reporte de todas las variantes de corte dependientes y de sustitución generadas, para una barra de acero en existencia.

- R25.** Emitir reporte de eficiencia y variables a contar en el modelo para una barra de acero en existencia.
- R26.** Emitir reporte de todas las variantes de corte generadas (primarias y de sustitución), para varias barras de acero en existencia.
- R27.** Emitir reporte de todas las variantes de corte primarias generadas, para varias barras de acero en existencia.
- R28.** Emitir reporte de todas las variantes de corte de sustitución generadas, para varias barras de acero en existencia.
- R29.** Emitir reporte de todas las variantes de corte independientes generadas (primarias y de sustitución), para varias barras de acero en existencia.
- R30.** Emitir reporte de todas las variantes de corte independientes primarias generadas, para varias barras de acero en existencia.
- R31.** Emitir reporte de todas las variantes de corte independientes y de sustitución generadas, para varias barras de acero en existencia.
- R32.** Emitir reporte de todas las variantes de corte dependientes generadas (primarias y de sustitución), para varias barras de acero en existencia.
- R33.** Emitir reporte de todas las variantes de corte dependientes primarias generadas, para varias barras de acero en existencia.
- R34.** Emitir reporte de todas las variantes de corte dependientes y de sustitución generadas, para varias barras de acero en existencia.
- R35.** Emitir reporte de eficiencia y variables a contar en el modelo para varias barras de acero en existencia.
- R36.** Emitir reporte del modelo matemático para una barra de acero en existencia.
- R37.** Emitir reporte del plan de corte optimizado para una barra de acero en existencia.
- R38.** Emitir reporte práctico del plan de corte optimizado para una barra de acero en existencia
- R39.** Emitir reporte del modelo matemático para varias barras de acero en existencia.
- R40.** Emitir reporte del plan de corte optimizado para varias barras de acero en existencia.
- R41.** Emitir reporte práctico del plan de corte optimizado para varias barras de acero en existencia.

R42. Generar fichero de las variantes de corte a tener en cuenta en el modelo para una barra de acero en existencia.

R43. Generar fichero de las variantes de corte a tener en cuenta en el modelo para varias barras de acero en existencia.

2.2.2.3 Requerimientos no funcionales.

Los requerimientos o requisitos no funcionales también conocidos como “requisitos adicionales”^[3] describen propiedades o cualidades que el producto debe tener tales como: “restricciones del entorno o de la implementación, rendimiento, dependencias de la plataforma, facilidad de mantenimiento, extensibilidad y fiabilidad (todas las ‘ades’).”^[4] Se refieren a las características que hacen al producto atractivo, usable, rápido o confiable y no a las funciones específicas que entrega el sistema.

De forma alternativa, definen las restricciones del sistema como la capacidad de los dispositivos de entrada/salida, en cuanto a prestaciones, atributos de calidad y la representación de datos que se utiliza en la interfaz del sistema.

Los requerimientos no funcionales del sistema propuesto son los siguientes:

Requerimientos de apariencia o interfaz externa.

- La interfaz se ajustará al estándar de ventanas que el sistema operativo Windows ha establecido y globalizado.
- La interfaz presentará un nivel de comprensión adecuado, respetando la uniformidad y la sencillez.
- La interfaz de la aplicación estará diseñada de modo tal que el usuario pueda tener en todo momento el control de la aplicación, lo que le permitirá ir de un punto a otro dentro de ella con gran facilidad. En la aplicación se posibilitará la visualización de varios reportes a la vez, con el fin de establecer comparaciones.

Requerimientos de uso:

- El software será utilizado por cualquier perfil de usuario sin importar si es o no un administrador.

^[3]Jacobson, Ivar. El Proceso Unificado de Desarrollo de Software (Tomo 1).--La Habana: Editorial Félix Varela, 2004.--p.125.

^[4]Jacobson, Ivar. El Proceso Unificado de Desarrollo de Software (Tomo 1).--La Habana: Editorial Félix Varela, 2004.--p.125.

- El sistema contará con una documentación básica que comprenda los aspectos generales para la utilización eficiente de la aplicación.
- El sistema podrá ser usado por personas que no tengan experiencia en el uso de la computadora, sólo necesitarían un ligero entrenamiento sobre el funcionamiento de los principales elementos de una interfaz estándar en el ambiente del sistema operativo Windows (uso del mouse, manejo de menús, botones, cuadros de texto).

Requerimientos de rendimiento:

- Los algoritmos (generar variantes de corte y Branch and Bound) supondrán un alto procesamiento puesto que se desarrollarán cálculos de gran envergadura⁵.
- Los tiempos de respuesta del sistema dependerán de las características de los datos que se estén procesando asegurando un alto nivel de confiabilidad.

Requerimientos de soporte:

- El sistema posibilitará su perfeccionamiento y la incorporación de nuevas opciones en un futuro.

Requerimientos de ayuda y documentación en línea:

- El sistema dispondrá de una ayuda bien detallada sobre las principales opciones que brinda, además de una documentación básica que comprenda los aspectos generales a tener en cuenta para trabajar con la aplicación.

Requerimiento de software

- Para la instalación del sistema propuesto se dispondrá del sistema operativo Windows 95 o versiones superiores.

Requerimiento de hardware

- Para la instalación del sistema propuesto se requerirá, como mínimo, de máquinas con las siguientes características:
 - Procesador de velocidad 1GHz.
 - 128 MB de RAM

⁵Estos problemas son denominados abiertos, es imposible conocer de antemano cuántas iteraciones se llevarán a cabo.

- 1 GB de HDD Libre

Restricciones en el diseño y la implementación

- Para garantizar una mejor documentación del sistema, se utilizará en el análisis, diseño e implementación el UML (Unified Modeling Language) y como herramienta de apoyo a este lenguaje de modelación se hará uso del Rational Rose.

Requerimientos de Seguridad

- Se deberá garantizar la integridad de los datos que se almacenen. La información almacenada será consistente y se utilizarán validaciones que limiten la entrada de datos incorrectos.
- La información estará disponible a los usuarios en todo momento, permitiendo un acceso rápido y seguro a la misma.

2.2.3 Modelo de casos de uso del sistema.

El modelo de casos de uso del sistema sirve de intérprete entre los desarrolladores y los clientes. “Permite que los desarrolladores de software y los clientes lleguen a un acuerdo sobre los requisitos. (...) El modelo de casos de uso sirve como acuerdo entre clientes y desarrolladores, y proporciona la entrada fundamental para el análisis, el diseño y las pruebas.”^[6]

El modelo de casos de uso del sistema describe a los actores del sistema, los casos de uso y sus relaciones, así como la descripción de cada uno de estos casos de uso.

2.2.3.1 Actores del sistema.

Los actores representan entes fuera del sistema que colaboran e interactúan con el mismo. “Es una entidad externa del sistema que de alguna manera participa en la historia del caso de uso”.^[7] “No todos los actores representan a personas. Pueden ser actores otros sistemas o hardware”.^[8] Un actor es aquel que interactúa con el sistema,

^[6]Jacobson, Ivar. El Proceso Unificado de Desarrollo de Software (Tomo 1).--La Habana: Editorial Félix Varela, 2004.--p.127.

^[7]Larman, Graig. UML y Patrones: Introducción al análisis y diseño orientado a objetos (Tomo 1).--La Habana: Editorial Félix Varela, 2004.--p.52.

^[8]Jacobson, Ivar. El Proceso Unificado de Desarrollo de Software (Tomo 1).--La Habana: Editorial Félix Varela, 2004.--p.127.

sin ser parte de él y puede asumir el rol que juega una o varias personas, un equipo o un sistema automatizado.

La comunicación actor-sistema se lleva a cabo mediante el envío o recepción de mensajes por parte de los actores utilizando para ello, los casos de uso. La función de un actor en esencia es accionar, “por lo regular estimula el sistema con eventos de entrada o recibe algo de él”.^[9] La definición clara de las responsabilidades de los actores y los casos de uso permite delimitar el alcance del sistema. Una vez identificados los actores del sistema, está definido el entorno externo del sistema.

Actor	Descripción
<p style="text-align: center;">Cliente</p>	<p>Personal que utiliza el sistema. Es el encargado de explotar todas las potencialidades que brinda el sistema.</p> <p>Requisitos funcionales asociados: R1, R2, R3, R4, R5, R6, R10, R13, R14, R15, R16, R17, R18, R19, R20, R21, R22, R23, R24, R25, R26, R27, R28, R29, R30, R31, R32, R33, R34, R35, R36, R37, R38, R39, R40, R41.</p>

Tabla 1. -Descripción del actor del sistema-

2.2.3.2 Casos de uso del sistema.

Según Jacobson, “un caso de uso especifica una secuencia de acciones, incluyendo variantes, que el sistema puede llevar a cabo, y que producen un resultado observable de valor para un actor concreto”.^[10] Por otra parte, Larman plantea que “los casos de uso son historias o casos de utilización de un sistema; no son exactamente los requerimientos ni las especificaciones funcionales, sino que ejemplifican e incluyen tácitamente los requerimientos en las historias que narran”.^[11] Su función consiste en darle cumplimiento a los deseos de los usuarios cuando utilizan el sistema.

^[9]Larman, Graig. UML y Patrones: Introducción al análisis y diseño orientado a objetos (Tomo 1).--La Habana: Editorial Félix Varela, 2004.--p.52.

^[10]Jacobson, Ivar. El Proceso Unificado de Desarrollo de Software (Tomo 1).--La Habana: Editorial Félix Varela, 2004.--p.39.

^[11]Larman, Graig. UML y Patrones: Introducción al análisis y diseño orientado a objetos (Tomo 1).--La Habana: Editorial Félix Varela, 2004.--p.49.

Los casos de uso son artefactos narrativos que describen, bajo la forma de acciones y reacciones, el comportamiento del sistema desde el punto de vista del usuario. De manera más precisa, un caso de uso especifica una secuencia de acciones que el sistema puede llevar a cabo interactuando con sus actores. Un modelo de casos de uso bien diseñado debe cuidar que capture todos los requisitos funcionales del sistema.

Los casos de uso para el sistema propuesto son los siguientes:

- CU1.** Introducir plan de corte para una materia prima en existencia.
- CU2.** Introducir plan de corte para varias materias prima en existencia.
- CU3.** Cargar plan de corte.
- CU4.** Guardar plan de corte.
- CU5.** Consultar ayuda.
- CU6.** Calcular variantes de corte para una materia prima en existencia.
- CU7.** Calcular variantes de corte para varias materias prima en existencia.
- CU8.** Ver reportes referentes a las variantes de corte para una materia prima en existencia.
- CU9.** Ver reportes referentes a las variantes de corte para varias materias prima en existencia.
- CU10.** Ver reporte de efectividad para una materia prima en existencia.
- CU11.** Ver reporte de efectividad para varias materias prima en existencia.
- CU12.** Calcular dependencias.
- CU13.** Suprimir variantes de corte dependientes.
- CU14.** Reporte de la forma canónica del modelo matemático para una materia prima en existencia.
- CU15.** Reporte de la forma canónica del modelo matemático para varias materias prima en existencia.
- CU16.** Reporte del plan de corte optimizado para una materia prima en existencia.
- CU17.** Reporte del plan de corte optimizado para varias materias prima en existencia.
- CU18.** Reporte práctico para una materia prima en existencia.
- CU19.** Reporte práctico para varias materias prima en existencia.
- CU20.** Optimizar modelo matemático.
- CU21.** Llevar modelo matemático de la forma canónica a la estándar.

CU22. Generar fichero, de las variantes de corte a tener en cuenta en el modelo, referente a una materia prima en existencia.

CU23. Generar fichero, de las variantes de corte a tener en cuenta en el modelo, referente a varias materias prima en existencia.

Donde la materia prima se define como barras de acero.

2.2.3.3 Diagrama de Casos de Uso del sistema.

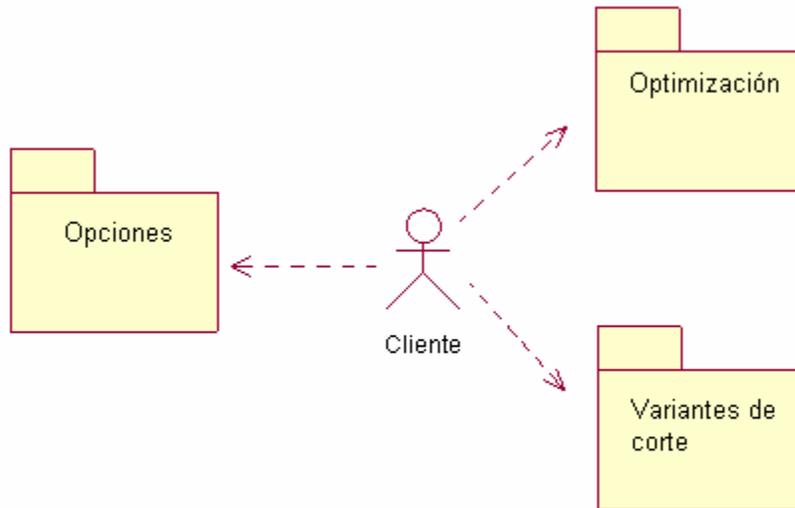


Ilustración 3. -Diagrama de casos de uso dividido por paquetes-

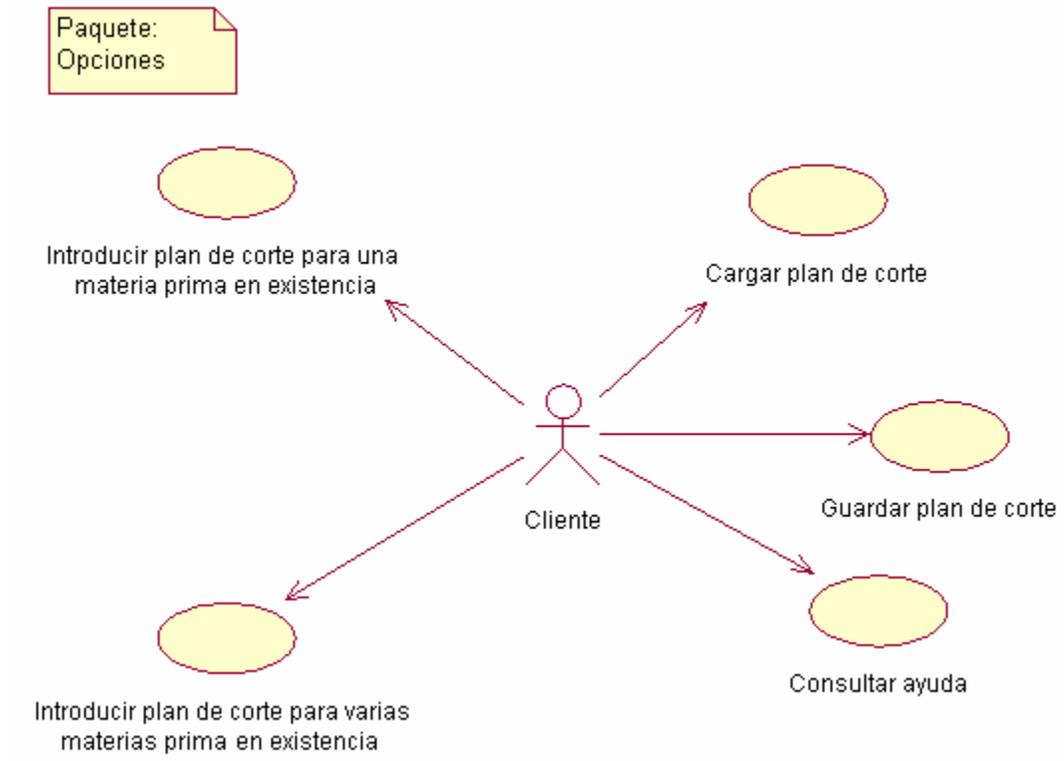


Ilustración 4. -Modelo de casos de uso. Paquete opciones-

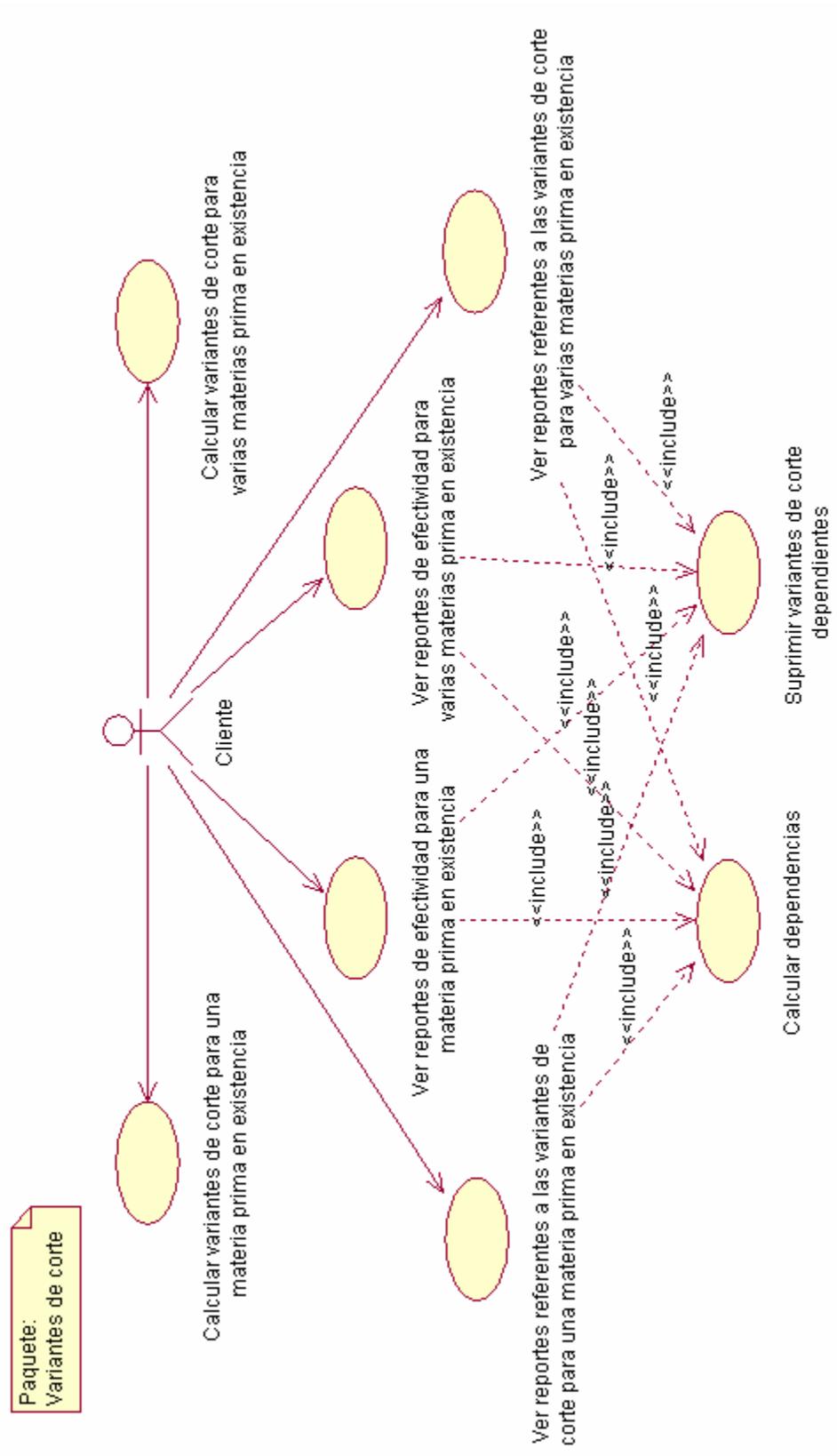


Ilustración 5. -Modelo de casos de uso. Paquetes variantes de corte-

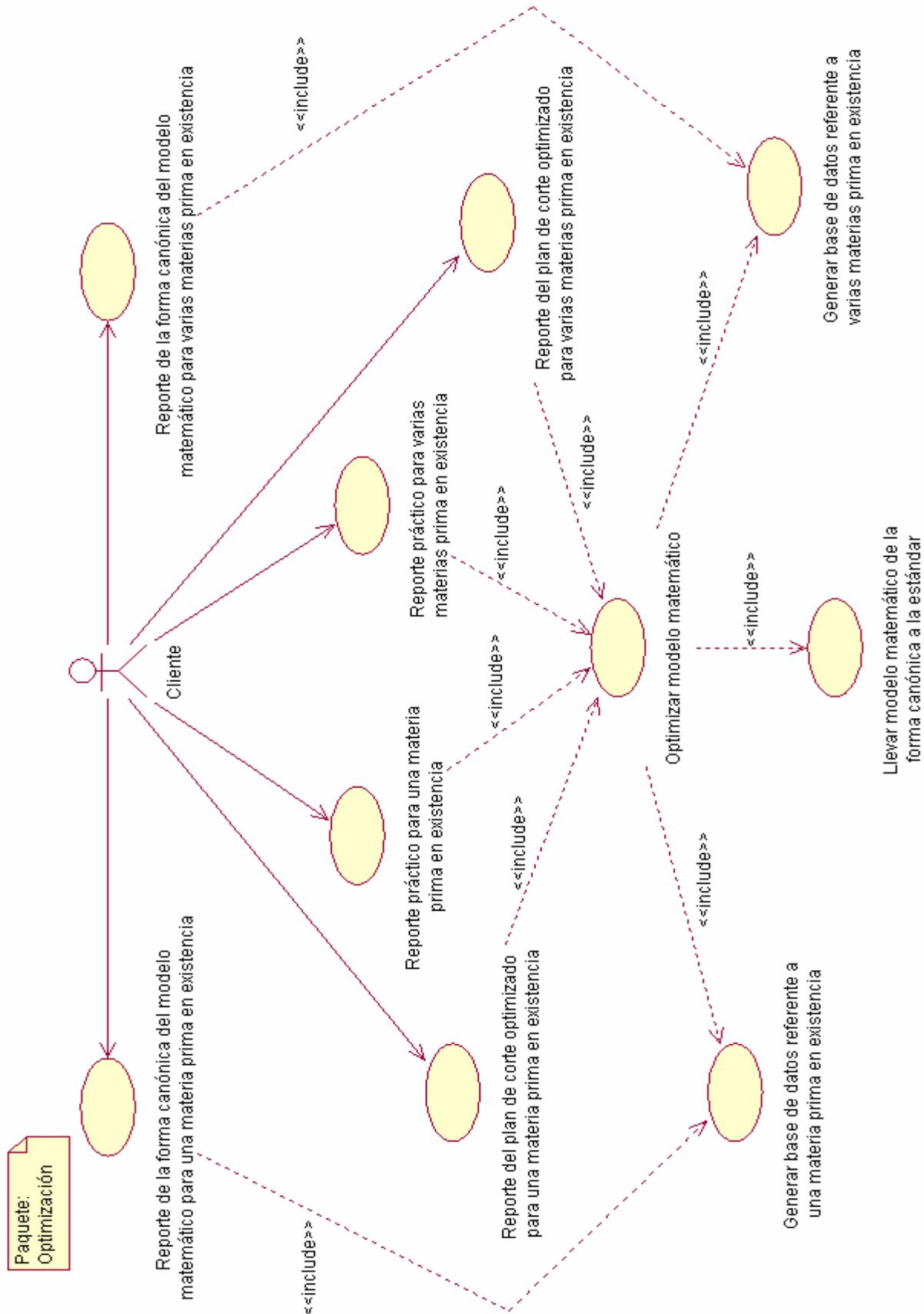


Ilustración 6. -Modelo de casos de uso. Paquetes optimización-

2.2.3.4 Descripción de los casos de uso del sistema.

Caso de uso 1.	Introducir plan de corte para una materia prima en existencia
Nombre	
Actor	Cliente.
Propósito	Introducir el plan de corte cuando en la entidad existe sólo un tipo de longitud de materia prima.
Resumen: El caso de uso se inicia cuando el cliente desee introducir los datos (plan de corte) teniendo en cuenta que existe sólo un tipo de longitud de materia prima. Luego, el sistema solicita la longitud de la materia prima con su cantidad en existencia, si constituye una restricción, y la longitud de cada pieza con la cantidad respectiva que se desee obtener. El caso de uso termina cuando el cliente ha introducido el plan de corte y el sistema ha aceptado los datos.	
Referencias	R1.
Precondiciones	El sistema debe haberse iniciado.
Post-condiciones	El sistema contiene el plan de corte referente a la existencia de un solo tipo de longitud de materia prima.
Prototipo	Anexo 4. Figura 5.

Tabla 2. -Caso de uso: Introducir plan de corte para una materia prima en existencia-

Caso de uso 2.	Introducir plan de corte para varias materias prima en existencia
Nombre	
Actor	Cliente.
Propósito	Introducir el plan de corte cuando en la entidad existen varias longitudes de materia prima.
Resumen: El caso de uso se inicia cuando el cliente desee introducir los datos (plan de corte) teniendo en cuenta que existen varias longitudes de materia prima. Luego, el sistema solicita las longitudes de la materia prima con las respectivas cantidades en existencia, si constituyen una restricción, y la longitud de cada pieza con la cantidad respectiva que se desee obtener. El caso de uso termina cuando el cliente ha	

introducido el plan de corte y el sistema ha aceptado los datos.	
Referencias	R2.
Precondiciones	El sistema debe haberse iniciado.
Post-condiciones	El sistema contiene el plan de corte referente a la existencia de varias longitudes de materia prima.
Prototipo	Anexo 4. Figura 6.

Tabla 3. -Caso de uso: Introducir plan de corte para varias materias prima en existencia-

Caso de uso 5.	Consultar ayuda
Nombre	
Actor	Cliente.
Propósito	Guiar al cliente en las distintas funciones que puede realizar en el sistema.
Resumen: El caso de uso se inicia cuando el cliente desee conocer alguna funcionalidad del sistema y recurre a visualizar la ayuda. Luego, el sistema ejecuta la acción de mostrar la ayuda. El caso de uso finaliza con la visualización de la ayuda.	
Referencias	R15.
Precondiciones	El sistema debe haberse iniciado.
Post-condiciones	--
Prototipo	Anexo 4. Figura 7.

Tabla 4. -Caso de uso: Consultar ayuda-

Caso de uso 4.	Guardar plan de corte.
Nombre	
Actor	Cliente.
Propósito	Guardar datos, tanto de una materia prima en existencia como de varias (plan de corte) en un archivo.

Resumen:	
<p>El caso de uso se inicia cuando el cliente desee guardar los datos que tiene en ese momento el sistema en un archivo, los cuales pueden ser referentes sólo a una longitud de materia prima en existencia, a varias o considerando ambos casos. Luego, el sistema solicita la ubicación donde se deseen guardar los datos. En caso en que ya se haya especificado la ubicación, el sistema guardará los nuevos cambios en la misma, sin solicitar ninguna información. El caso de uso termina cuando el cliente ha guardado los datos.</p>	
Referencias	R13.
Precondiciones	El sistema debe haberse iniciado y debe contar con datos.
Post-condiciones	--
Prototipo	Anexo 4. Figura 8.

Tabla 5. -Caso de uso: Guardar plan de corte-

Caso de uso 3.	Cargar plan de corte.
Nombre	
Actor	Cliente.
Propósito	Obtener de un archivo datos existentes, tanto de una materia prima en existencia como de varias (plan de corte).
Resumen:	
<p>El caso de uso se inicia cuando el cliente desee abrir un archivo que contenga datos de un plan de corte, pueden ser referentes sólo a una longitud de materia prima en existencia, a varias o considerando ambos casos. Luego, se solicita la ubicación del archivo a ser abierto.</p> <p>En el caso que se entren nuevos datos referentes a un plan de corte o se modifiquen los existentes, antes de solicitar la ubicación se da la posibilidad de guardarlos. El caso de uso termina cuando el cliente ha introducido la ubicación del archivo y el sistema ha aceptado los datos.</p>	
Referencias	R14.

Precondiciones	El sistema debe haberse iniciado.
Post-condiciones	El sistema contiene el nuevo plan de corte cargado.
Prototipo	Anexo 4. Figura 9.

Tabla 6. -Caso de uso: Cargar plan de corte-

Caso de uso 6.	Calcular variantes de corte para una materia prima en existencia
Nombre	
Actor	Cliente.
Propósito	Calcular las variantes de corte primarias y de sustitución tomando en cuenta un sólo tipo de longitud de materia prima.
Resumen:	
<p>El caso de uso se inicia cuando el cliente desee visualizar el reporte con todas las variantes de corte primarias y de sustitución considerando sólo un tipo de longitud de materia prima. Luego, se ejecuta la acción de generar las variantes de corte. Esta acción se lleva a cabo únicamente la primera vez que el cliente desee visualizar el reporte, sólo se volverá a ejecutar si hay cambios en los datos o se entran nuevos datos. Internamente, el sistema buscará todas las combinaciones posibles de corte de la materia prima para obtener las piezas (primarias) y todas las combinaciones de corte entre las mismas piezas (de sustitución). El caso de uso termina con la visualización del reporte.</p>	
Referencias	R3, R4, R16.
Precondiciones	El sistema debe haberse iniciado y contar con datos referentes a un sólo tipo de longitud de materia prima.
Post-condiciones	El sistema contiene las variantes de corte primarias y de sustitución referentes a un sólo tipo de longitud de materia prima.
Prototipo	Anexo 4. Figura 10.

Tabla 7. -Caso de uso: Calcular variantes de corte para una materia prima en existencia-

Caso de uso 7.	Calcular variantes de corte para varias materias prima en existencia
Nombre	
Actor	Cliente.
Propósito	Calcular las variantes de corte primarias y de sustitución tomando en cuenta varias longitudes de materia prima.
Resumen: El caso de uso se inicia cuando el cliente desee visualizar el reporte con todas las variantes de corte primarias y de sustitución considerando varias longitudes de materia prima. Luego, se ejecuta la acción de generar las variantes de corte. Esta acción se lleva a cabo únicamente la primera vez que el cliente desee visualizar el reporte, sólo se volverá a ejecutar si hay cambios en los datos o se entran nuevos datos. Internamente, el sistema buscará todas las combinaciones posibles de corte de las materias prima para obtener las piezas (primarias) y todas las combinaciones de corte entre las mismas piezas (de sustitución). El caso de uso termina con la visualización del reporte.	
Referencias	R5, R6, R26.
Precondiciones	El sistema debe haberse iniciado y contar con datos referente a varias longitudes de materia prima.
Post-condiciones	El sistema contiene las variantes de corte primarias y de sustitución referentes a varias longitudes de materia prima.
Prototipo	Anexo 4. Figura 11.

Tabla 8. -Caso de uso: Calcular variantes de corte para varias materias prima en existencia-

Caso de uso 8.	Ver reporte referente a las variantes de corte para una materia prima en existencia
Nombre	
Actor	Cliente.
Propósito	Visualizar una serie de reportes de importancia vinculados con las variantes de corte considerando sólo un tipo de longitud de materia prima.

Resumen:

El caso de uso se inicia cuando el cliente desee visualizar cualquiera de los reportes vinculados con las variantes de corte considerando sólo un tipo de longitud de materia prima. Ellos son los siguientes:

- Reporte de todas las variantes de corte primarias generadas. Muestra, de todas las variantes de corte, aquellas que son primarias.
- Reporte de todas las variantes de corte de sustitución generadas. Muestra, de todas las variantes de corte, aquellas que son de sustitución.
- Reporte de todas las variantes de corte independientes generadas. Muestra, de todas las variantes de corte, primarias y de sustitución, aquellas que son independientes.
- Reporte de todas las variantes de corte independientes primarias generadas. Muestra, de todas las variantes de corte independientes, aquellas que son primarias.
- Reporte de todas las variantes de corte independientes que son de sustitución generadas. Muestra, de todas las variantes de corte independientes, aquellas que son de sustitución.
- Reporte de todas las variantes de corte dependientes generadas. Muestra, de todas las variantes de corte, primarias y de sustitución, aquellas que son dependiente.
- Reporte de todas las variantes de corte dependientes primarias generadas. Muestra, de todas las variantes de corte dependientes, las que son primarias.
- Reporte de todas las variantes de corte dependientes y de sustitución generadas. Muestra, de todas las variantes de corte dependientes, las que son de sustitución.

Luego, se ejecuta la acción de buscar dependencias. Esta acción se lleva a cabo únicamente la primera vez que el cliente desee visualizar uno de estos reportes, sólo se volverá a ejecutar si hay cambios en los datos o se entran nuevos datos. Seguidamente se realiza la acción de suprimir las variantes de corte dependientes siempre y cuando el reporte lo necesite. Internamente, el sistema buscará aquellas variantes de corte que se puedan escribir como combinación lineal de otras dos y las

marcará, mostrándolas o no en ocasiones. El caso de uso termina con la visualización del reporte seleccionado.	
Referencias	R17, R18, R19, R20, R21, R22, R23, R24, CU12(include), CU13(include).
Precondiciones	El sistema debe haberse iniciado y contar con las variantes de corte primarias y de sustitución referentes a un solo tipo de longitud de materia prima.
Post-condiciones	--
Prototipo	Anexo 4. Figura 12.

Tabla 9. -Caso de uso: Ver reporte referente a las variantes de corte para una materia prima en existencia-

Caso de uso 9.	Ver reporte referente a las variantes de corte para varias materias prima en existencia
Nombre	
Actor	Cliente.
Propósito	Visualizar una serie de reportes de importancia vinculados con las variantes de corte considerando varias longitudes de materia prima.
Resumen:	
<p>El caso de uso se inicia cuando el cliente desee visualizar cualquiera de los reportes vinculados con las variantes de corte considerando varias longitudes de materia prima. Ellos son los siguientes:</p> <ul style="list-style-type: none"> • Reporte de todas las variantes de corte primarias generadas. Muestra, de todas las variantes de corte, aquellas que son primarias. • Reporte de todas las variantes de corte de sustitución generadas. Muestra, de todas las variantes de corte, aquellas que son de sustitución. • Reporte de todas las variantes de corte independientes generadas. Muestra, de todas las variantes de corte, primarias y de sustitución, aquellas que son independiente. • Reporte de todas las variantes de corte independientes primarias generadas. Muestra, de todas las variantes de corte independientes, 	

aquellas que son primarias.

- Reporte de todas las variantes de corte independientes y de sustitución generadas. Muestra, de todas las variantes de corte independientes, aquellas que son de sustitución.
- Reporte de todas las variantes de corte dependientes generadas. Muestra, todas las variantes de corte, primarias y de sustitución, aquellas que son dependiente.
- Reporte de todas las variantes de corte dependientes y primarias generadas. Muestra, de todas las variantes de corte dependientes, las que son primarias.
- Reporte de todas las variantes de corte dependientes y de sustitución generadas. Muestra, de todas las variantes de corte dependientes, las que son de sustitución.

Luego, se ejecuta la acción de buscar dependencias. Esta acción se lleva a cabo únicamente la primera vez que el cliente desee visualizar uno de estos reportes, sólo se volverá a ejecutar si hay cambios en los datos o se entran nuevos datos. Seguidamente se realiza la acción de suprimir las variantes de corte dependientes siempre y cuando el reporte lo necesite. Internamente, el sistema buscará aquellas variantes de corte que se puedan escribir como combinación lineal de otras dos y las marcará, mostrándolas o no en ocasiones. El caso de uso termina con la visualización del reporte seleccionado.

Referencias	R27, R28, R29, R30, R31, R32, R33, R34, CU12(include), CU13(include).
Precondiciones	El sistema debe haberse iniciado y contar con las variantes de corte primarias y de sustitución referentes a varias longitudes de materia prima.
Post-condiciones	--
Prototipo	Anexo 4. Figura 13.

Tabla 10. -Caso de uso: Ver reporte referente a las variantes de corte para varias materias prima en existencia-

Caso de uso 10.	Ver reporte de efectividad para una materia prima en existencia
Nombre	
Actor	Cliente.
Propósito	Visualizar reporte del por ciento de efectividad de reducción de las variantes de corte primarias, considerando un solo tipo de longitud de materia prima.
Resumen: El caso de uso se inicia cuando el cliente desee visualizar el reporte de por ciento de efectividad de reducción de las variantes de corte primarias, considerando un solo tipo de longitud de materia prima. Luego, y si no han sido buscadas anteriormente por otros casos de uso, se ejecuta la acción de buscar dependencias. Esta acción se lleva a cabo únicamente la primera vez que el cliente desee visualizar este reporte, sólo se volverá a ejecutar si hay cambios en los datos o se entran nuevos datos. Seguidamente se realiza la acción de suprimir las variantes de corte dependientes. Internamente, el sistema calculará el por ciento entre las variantes de corte primarias y todas las que resulten independientes. El caso de uso termina con la visualización del reporte.	
Referencias	R25, CU12(include), CU13(include).
Precondiciones	El sistema debe haberse iniciado y contar con las variantes de corte primarias y de sustitución referentes a un solo tipo de longitud de materia prima.
Post-condiciones	--
Prototipo	Anexo 4. Figura 14.

Tabla 11. -Caso de uso: Ver reporte de efectividad para una materia prima en existencia-

Caso de uso 11.	Ver reporte de efectividad para varias materias prima en existencia
Nombre	
Actor	Cliente.
Propósito	Visualizar reporte del por ciento de efectividad de reducción de las variantes de corte primarias, considerando varias longitudes de materia prima.

Resumen:	
<p>El caso de uso se inicia cuando el cliente desee visualizar el reporte de por ciento de efectividad de reducción de las variantes de corte primarias, considerando varias longitudes de materia prima. Luego, y si no han sido buscadas anteriormente por otros casos de uso, se ejecuta la acción de buscar dependencias. Esta acción se lleva a cabo únicamente la primera vez que el cliente desee visualizar este reporte, sólo se volverá a ejecutar si hay cambios en los datos o se entran nuevos datos. Seguidamente se realiza la acción de suprimir las variantes de corte dependientes. Internamente, el sistema calculará el por ciento entre las variantes de corte primarias y todas las que resulten independientes. El caso de uso termina con la visualización del reporte.</p>	
Referencias	R35, CU12(include), CU13(include).
Precondiciones	El sistema debe haberse iniciado y contar con las variantes de corte primarias y de sustitución referentes a varias longitudes de materia prima.
Post-condiciones	--
Prototipo	Anexo 4. Figura 15.

Tabla 12. -Caso de uso: Ver reporte de efectividad para varias materias prima en existencia-

Caso de uso 12.	Calcular dependencias
Nombre	
Actor	Cliente.
Propósito	Calcular todas las variantes de corte que resulten de la combinación de otras dos.
Resumen:	
<p>El caso de uso se inicia mediante otro caso de uso, que incluye a este como parte de su funcionamiento. Luego, comienza el cálculo de dependencias. Internamente, el sistema recorre cada variante de corte y tomando dos a dos verifica las combinaciones que puedan existir entre ellas. La búsqueda de variantes de corte dependientes primarias se realiza entre las variantes de corte primarias y las de sustitución, mientras que la búsqueda de variantes de corte dependientes que sean de</p>	

sustitución se realiza entre las variantes de corte de sustitución. El caso de uso finaliza cuando se han calculado las dependencias entre las variantes de corte.	
Referencias	R7, R8.
Precondiciones	El sistema debe haberse iniciado y contar con todas las variantes de corte sin importar la cantidad de tipos de longitudes de materia prima que existan.
Post-condiciones	El sistema contiene las dependencias entre las variantes de corte.
Prototipo	--

Tabla 13. -Caso de uso: Calcular dependencias-

Caso de uso 13.	Suprimir variantes de corte dependientes
Nombre	
Actor	Cliente.
Propósito	Suprimir aquellas variantes de corte que se puedan escribir como combinación lineal de otras dos.
Resumen: El caso de uso se inicia mediante otro caso de uso, que incluye a este como parte de su funcionamiento. Luego, se procede a suprimir aquellas variantes de corte dependientes. Internamente, el sistema va recorriendo cada variante de corte y al detectar que esta es dependiente, la marca. El caso de uso finaliza cuando se han suprimido todas las variantes de corte dependientes.	
Referencias	R9.
Precondiciones	El sistema debe contar con todas las variantes de corte y con las dependencias calculadas.
Post-condiciones	--
Prototipo	--

Tabla 14. -Caso de uso: Suprimir variantes de corte dependientes-

Caso de uso 14.	Reporte de la forma canónica del modelo matemático para una materia prima en existencia
Nombre	
Actor	Cliente.
Propósito	Visualizar el modelo matemático considerando el plan de corte referente a un solo tipo de longitud de materia prima.
Resumen: El caso de uso se inicia cuando el cliente desee visualizar el modelo matemático considerando el plan de corte referente a un solo tipo de longitud de materia prima. Luego, el sistema utiliza su base de datos para elaborar la forma canónica del modelo matemático (tal y como se concibe el modelo) visualizando un reporte con el mismo. El reporte debe especificar para cada variable qué variante de corte le corresponde. Si la base de datos no está creada, se crea antes de elaborar el modelo matemático. El caso de uso termina con la visualización del reporte.	
Referencias	R10, R36, CU22 (include).
Precondiciones	El sistema debe haberse iniciado y contar con datos referentes a un solo tipo de longitud de materia prima.
Post-condiciones	--
Prototipo	Anexo 4. Figura 16.

Tabla 15. -Caso de uso: Reporte de la forma canónica del modelo matemático para una materia prima en existencia-

Caso de uso 15.	Reporte de la forma canónica del modelo matemático para varias materias prima en existencia
Nombre	
Actor	Cliente.
Propósito	Visualizar el modelo matemático considerando el plan de corte referente a varios tipos de longitudes de materia prima.
Resumen: El caso de uso se inicia cuando el cliente desee visualizar el modelo matemático	

<p>considerando varios tipos de longitudes de materia prima. Luego, el sistema utiliza su base de datos para elaborar la forma canónica del modelo matemático (tal y como se concibe el modelo) visualizando un reporte con el mismo. El reporte debe especificar para cada variable qué variante de corte le corresponde. Si la base de datos no está creada, se crea antes de elaborar el modelo matemático. El caso de uso termina con la visualización del reporte.</p>	
Referencias	R10, R39, CU23 (include).
Precondiciones	El sistema debe haberse iniciado y contar con datos referentes a varios tipos de longitudes de materia prima.
Post-condiciones	--
Prototipo	Anexo 4. Figura 17.

Tabla 16. -Caso de uso: Reporte de la forma canónica del modelo matemático para varias materias prima en existencia-

Caso de uso 16.	Reporte del plan de corte optimizado para una materia prima en existencia
Nombre	
Actor	Cliente.
Propósito	Visualizar reporte con el plan de corte optimizado (solución matemática) referente a un solo tipo de longitud de materia prima.
Resumen:	
<p>El caso de uso se inicia cuando el cliente desee visualizar el reporte con el plan de corte optimizado (solución matemática) referente a un solo tipo de longitud de materia prima. Luego, se pasa a buscar la solución óptima. El caso de uso finaliza con la visualización de la solución óptima del problema.</p>	
Referencias	R37, CU20 (include), CU21 (include).
Precondiciones	El sistema debe haberse iniciado y contar con datos referentes a un solo tipo de longitud de materia prima.
Post-condiciones	--
Prototipo	Anexo 4. Figura 18.

Tabla 17. -Caso de uso: Reporte del plan de corte optimizado para una materia prima en existencia-

Caso de uso 17.	Reporte del plan de corte optimizado para varias materias prima en existencia
Nombre	
Actor	Cliente.
Propósito	Visualizar reporte con el plan de corte optimizado (solución matemática) referente a varios tipos de longitudes de materia prima.
Resumen: El caso de uso se inicia cuando el cliente desee visualizar el reporte con el plan de corte optimizado (solución matemática) referente a varios tipos de longitudes de materia prima. Luego, se pasa a buscar la solución óptima. El caso de uso finaliza con la visualización de la solución óptima del problema.	
Referencias	R40, CU20 (include), CU21 (include).
Precondiciones	El sistema debe haberse iniciado y contar con datos referentes a varios tipos de longitudes de materia prima.
Post-condiciones	--
Prototipo	Anexo 4. Figura 19.

Tabla 18. -Caso de uso: Reporte del plan de corte optimizado para varias materias prima en existencia-

Caso de uso 18.	Reporte práctico para una materia prima en existencia
Nombre	
Actor	Cliente.
Propósito	Visualizar reporte práctico referente a un solo tipo de longitud de materia prima.
Resumen: El caso de uso se inicia cuando el cliente desee visualizar el reporte práctico referente a un solo tipo de longitud de materia prima. Luego, se pasa a buscar la solución óptima. Este reporte consiste, en que de una forma que sea entendible por cualquier cliente, se exponga el plan de corte a llevar a cabo garantizando que el desperdicio de materia prima sea el mínimo (interpretación de los resultados). El caso de uso finaliza con la visualización del reporte.	

Referencias	R38, CU20(include), CU21(include).
Precondiciones	El sistema debe haberse iniciado y contar con datos referente a un solo tipo de longitud de materia prima.
Post-condiciones	--
Prototipo	Anexo 4. Figura 20.

Tabla 19. -Caso de uso: Reporte práctico para una materia prima en existencia-

Caso de uso 19.	Reporte práctico para varias materias prima en existencia
Nombre	
Actor	Cliente.
Propósito	Visualizar reporte práctico referente a varios tipos de longitudes de materia prima.
Resumen: El caso de uso se inicia cuando el cliente desee visualizar el reporte práctico referente a varios tipos de longitudes de materia prima. Luego, se pasa a buscar la solución óptima. Este reporte consiste, en que de una forma que sea entendible por cualquier cliente, se exponga el plan de corte a llevar a cabo garantizando que el desperdicio de materia prima sea el mínimo (interpretación de los resultados). El caso de uso finaliza con la visualización del reporte.	
Referencias	R41, CU20(include), CU21(include).
Precondiciones	El sistema debe haberse iniciado y contar con datos referentes a varios tipos de longitudes de materia prima.
Post-condiciones	--
Prototipo	Anexo 4. Figura 21.

Tabla 20. -Caso de uso: Reporte práctico para varias materias prima en existencia-

Caso de uso 20.	Optimizar modelo matemático
Nombre	
Actor	Cliente.
Propósito	Realizar el cálculo para obtener la solución óptima

	entera del modelo matemático.
Resumen:	El caso de uso se inicia mediante otro caso de uso, que incluye a este como parte de su funcionamiento. Luego, el sistema utiliza su base de datos para elaborar la forma canónica del modelo matemático, llevarlo de la forma canónica a una en que pueda ser procesado (estándar) y resolverlo para encontrar la solución óptima. Internamente, el sistema aplica el método de ramas y cotas (Branch and Bound) para la búsqueda de soluciones enteras. Si la base de datos no está creada, se crea antes de llevar a cabo este proceso. El caso de uso termina con el cálculo de la solución óptima.
Referencias	R10, R12, CU21 (include), CU22 (include), CU23 (include).
Precondiciones	El sistema debe haberse iniciado y contar con el plan de corte correspondiente.
Post-condiciones	El sistema contiene la solución óptima del problema.
Prototipo	--

Tabla 21. -Caso de uso: Optimizar modelo matemático-

Caso de uso 21.	Llevar modelo matemático de la forma canónica a la estándar
Nombre	
Actor	Cliente.
Propósito	Transformar el modelo matemático tal y como es definido (forma canónica) a una forma que pueda ser procesado (estándar).
Resumen:	El caso de uso es iniciado por el caso de uso <i>Optimizar modelo matemático</i> . Luego, convierte el modelo matemático de la forma canónica a la estándar. Internamente, el sistema transforma el modelo matemático de forma tal que quede expresado solamente en ecuaciones. El caso de uso termina con la transformación del modelo matemático.
Referencias	R11.
Precondiciones	El sistema debe haberse iniciado y contar con el plan de

	corte correspondiente.
Post-condiciones	El sistema contiene el modelo matemático en la forma estándar.
Prototipo	--

Tabla 22. -Caso de uso: Llevar modelo matemático de la forma canónica a la estándar-

Caso de uso 22.	Generar fichero, de las variantes de corte a tener en cuenta en el modelo, referente a una materia prima en existencia
Nombre	
Actor	Cliente.
Propósito	Liberar memoria que era ocupada por las variantes de corte referentes a un solo tipo de longitud de materia prima en existencia y que pudiera ser requerida en el proceso de optimización.
Resumen: El caso de uso se inicia mediante otro caso de uso. Luego, el sistema genera una base datos con las variantes de corte referentes a un solo tipo de longitud de materia prima en existencia. Si no se han calculado las variantes de corte y las dependencias, el flujo de trabajo se pasa a estos casos de uso retornando luego a este. Internamente, el sistema almacena: el plan de corte, las variables a considerar en el modelo, la relación variables variantes de corte y las variantes de corte. El caso de uso finaliza con la creación de la base de datos referente a un solo tipo de longitud de materia prima en existencia.	
Referencias	R42.
Precondiciones	El sistema debe haberse iniciado y contar con el plan de corte correspondiente a un solo tipo de longitud de materia prima en existencia.
Post-condiciones	El sistema ha creado la base de datos referentes a un solo tipo de longitud de materia prima.
Prototipo	--

Tabla 23. -Caso de uso: Generar bases de datos referentes a una materia prima en existencia-

Caso de uso 23.	Generar fichero, de las variantes de corte a tener en cuenta en el modelo, referente a varias materias prima en existencia
Nombre	
Actor	Cliente.
Propósito	Liberar memoria que era ocupada por las variantes de corte referentes a varios tipos de longitudes de materia prima en existencia y que pudiera ser requerida en el proceso de optimización.
Resumen:	
<p>El caso de uso se inicia mediante otro caso de uso. Luego, el sistema genera una base datos con las variantes de corte referentes a varios tipos de longitudes de materia prima en existencia. Si no se han calculado las variantes de corte y las dependencias, el flujo de trabajo se pasa a estos casos de uso retornando luego a este. Internamente, el sistema almacena: el plan de corte, las variables a considerar en el modelo, la relación variables variantes de corte y las variantes de corte. El caso de uso finaliza con la creación de la base de datos referente a varios tipos de longitudes de materia prima en existencia.</p>	
Referencias	R43.
Precondiciones	El sistema debe haberse iniciado y contar con el plan de corte correspondiente a varios tipos de longitudes de materia prima en existencia.
Post-condiciones	El sistema ha creado la base de datos referente a varios tipos de longitudes de materia prima.
Prototipo	--

Tabla 24. -Caso de uso: Generar bases de datos referentes a varias materias prima en existencia-

2.3 Construcción de la solución propuesta.

2.3.1 Diagrama de clases del diseño.

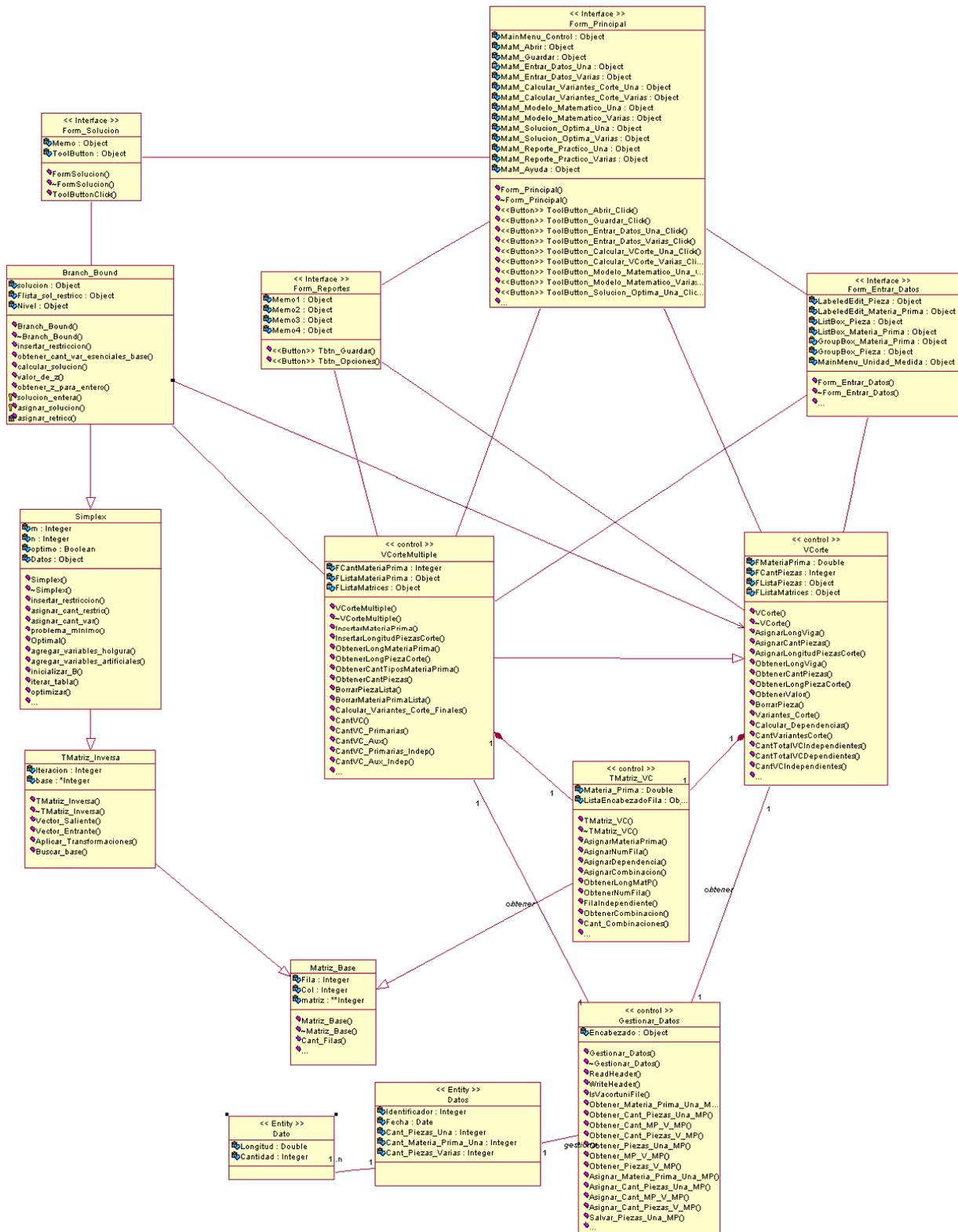


Ilustración 7. -Diagrama de clases del diseño-

A continuación, y para mejor visualización, se presenta el diagrama en tres partes:

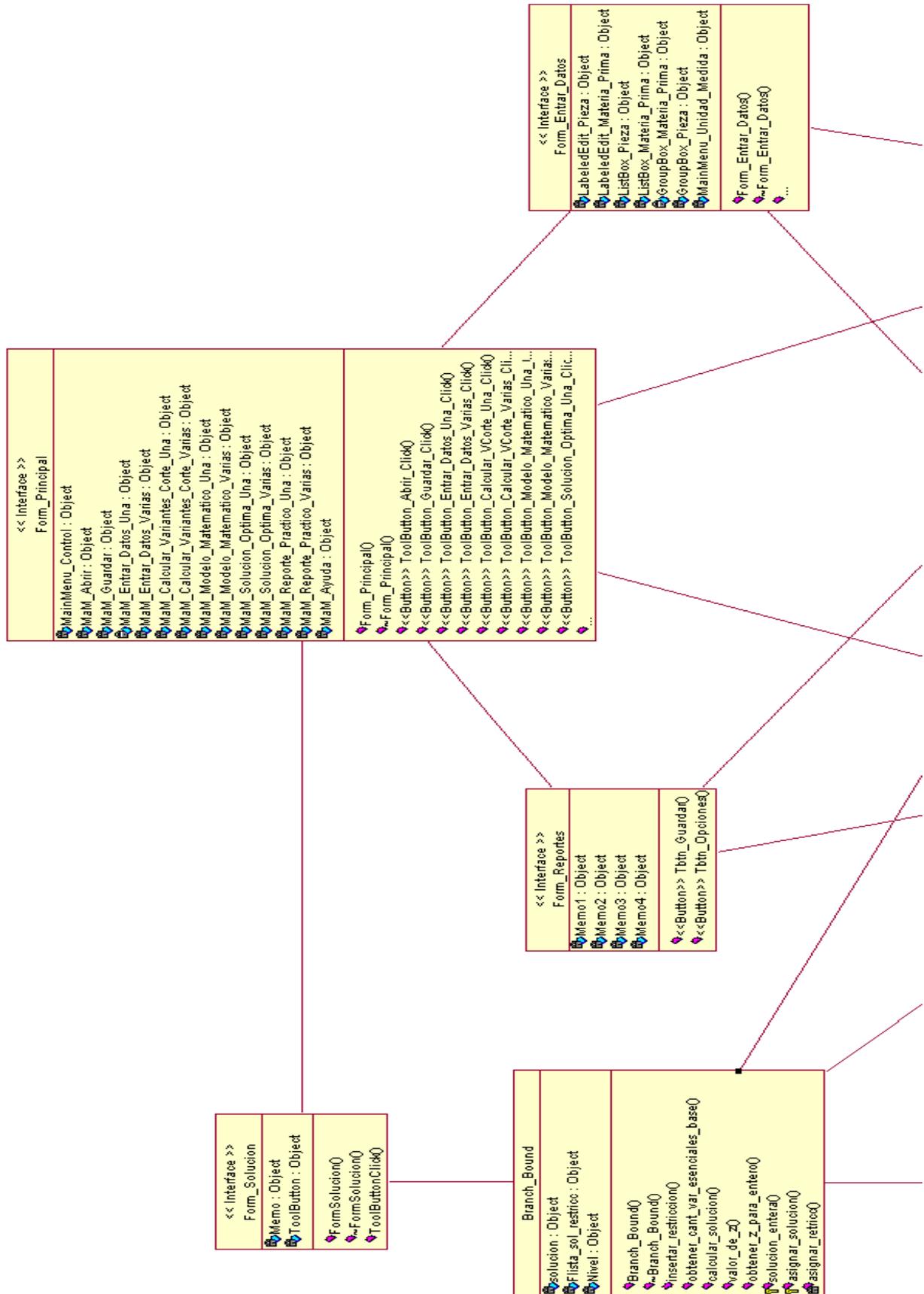


Ilustración 8. -Diagrama de clases del diseño. Parte 1-

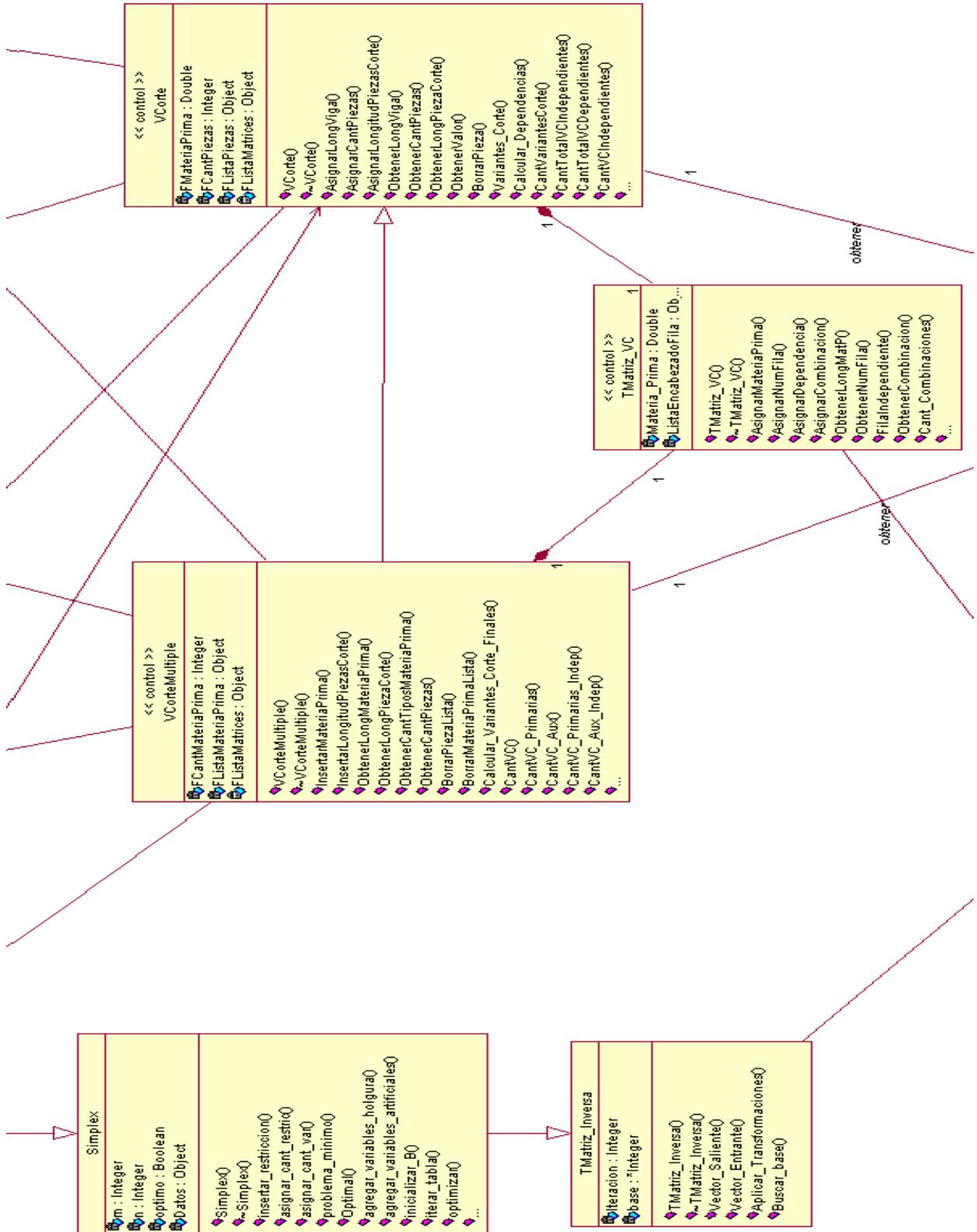


Ilustración 9. -Diagrama de clases del diseño. Parte 2-

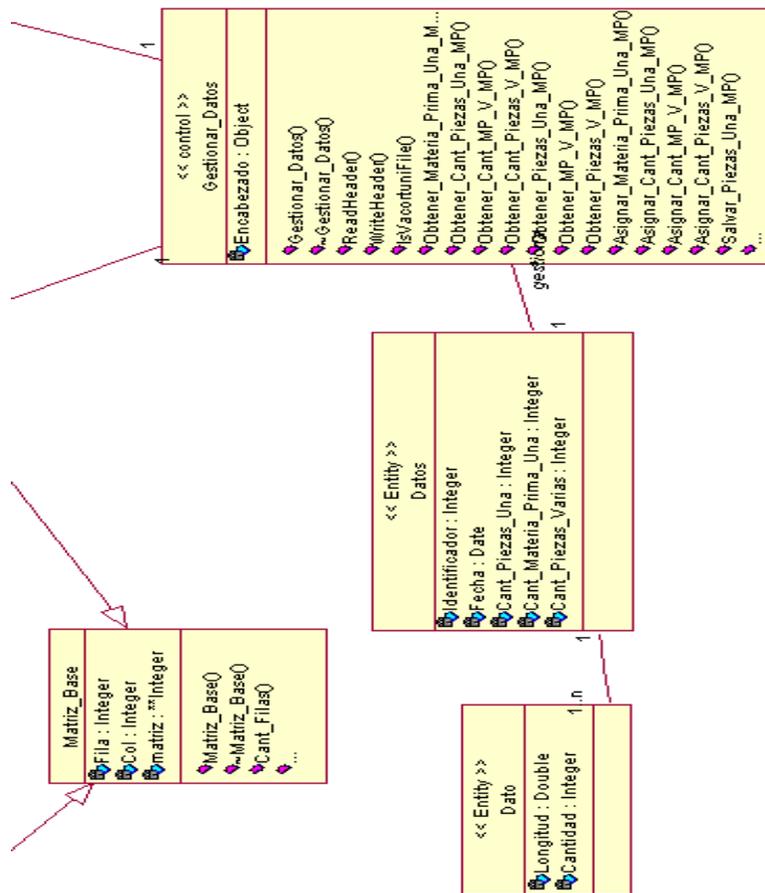


Ilustración 10. -Diagrama de clases del diseño. Parte 3-

2.3.2 Diseño de archivos.

2.3.2.1 Diagrama de archivo.

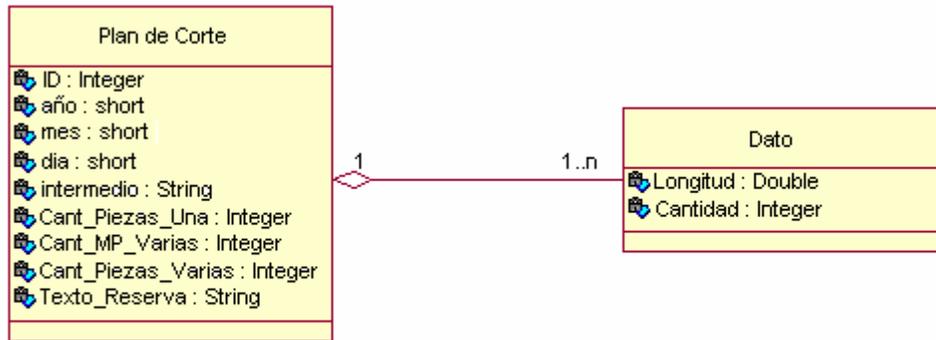


Ilustración 11. -Diagrama de clases persistentes. Plan de corte-

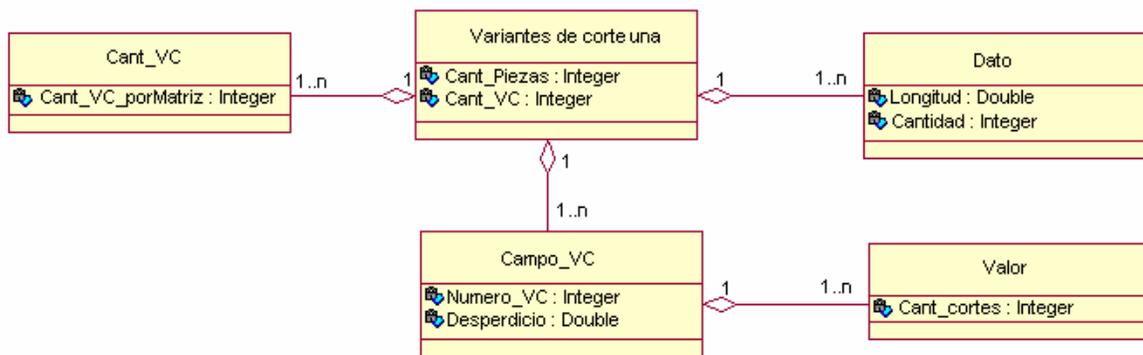


Ilustración 12. -Diagrama de clases persistentes. Variantes de corte para una materia prima en existencia-

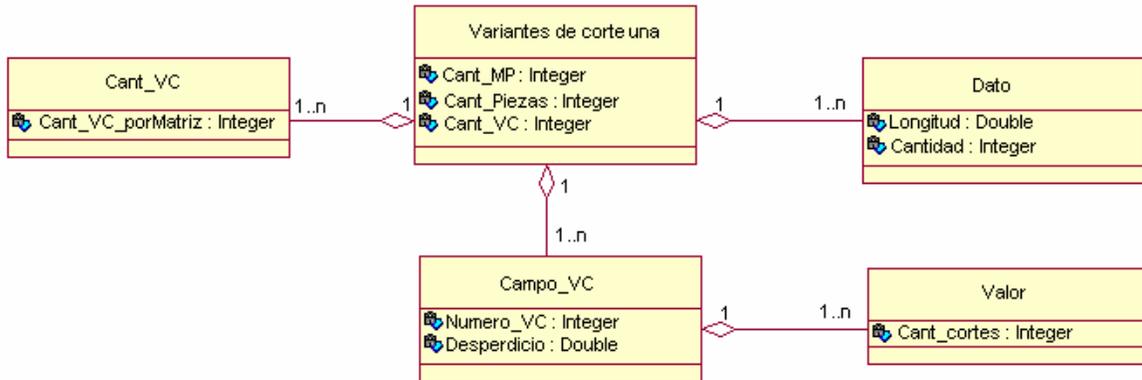


Ilustración 13. -Diagrama de clases persistentes. Variantes de corte para varias materias prima en existencia-

2.3.2.2 Estructura del archivo.

Archivo plan de corte:



Ilustración 14. -Estructura del archivo. Plan de corte-

A continuación se describe cada campo del archivo:

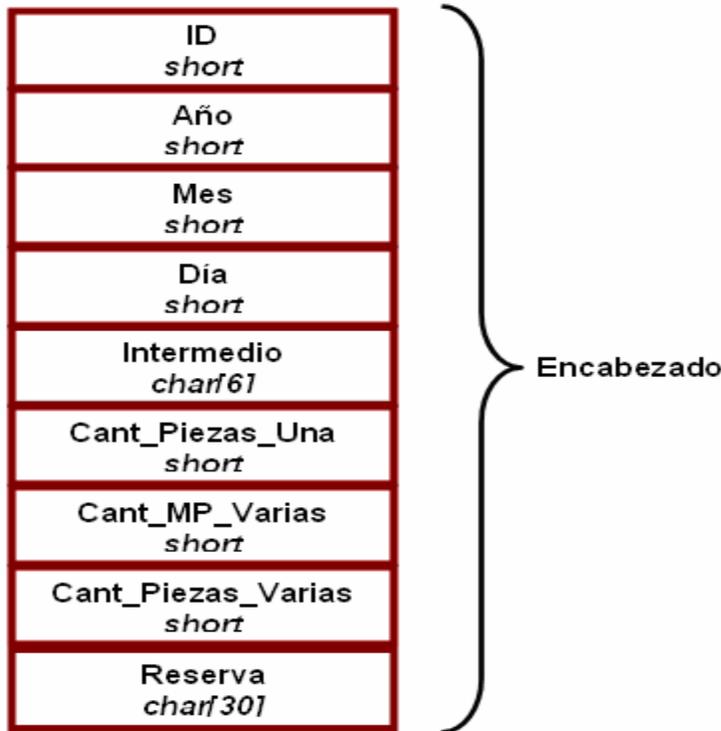


Ilustración 15. -Estructura del archivo.
Encabezado de plan de corte-

Descripción de los campos del encabezado:

ID: Identifica que el archivo fue creado por la aplicación.

Año: Año en que fue creado o modificado el archivo.

Mes: Mes en que fue creado o modificado el archivo.

Día: Día en que fue creado o modificado el archivo.

Intermedio: Espacio a utilizar opcionalmente.

Cant_Piezas_Una: Cantidad de piezas referente a un solo tipo de longitud de materia prima.

Cant_MP_Varias: Cantidad de tipos de longitud de materia prima.

Cant_Piezas_Varias: Cantidad de piezas referente a varios tipos de longitud de materia prima.



Ilustración 16. -Estructura del archivo. Dato de plan de corte-

Intermedio: Espacio a utilizar opcionalmente.

Descripción de los campos de datos:

Longitud: Longitud de materia prima o pieza.

Cantidad: Cantidad en existencia o que se desea

Archivo de las variantes de corte a tener en cuenta en el modelo para una longitud de materia prima en existencia:



Ilustración 17. -Estructura del archivo. Variantes de corte para una materia prima-
A continuación se describe cada campo del archivo:



Ilustración 18. -Estructura del archivo. Encabezado de variantes de corte para una materia prima-

Descripción de los campos del encabezado:

Cantidad_Piezas: Cantidad de piezas para un solo tipo de longitud de materia prima.

Cantidad_VC: Cantidad de variantes de corte a considerar en el modelo matemático.



Ilustración 19. -Estructura del archivo. Cantidad de variantes de corte del archivo variantes de corte para una materia prima-

Descripción del campo de VC por matrices:

Cant_VC_porMatriz: Cantidad de variantes de corte que tiene cada matriz. Las matrices se refieren a variantes de corte primarias o de sustitución.



Ilustración 20. -Estructura del archivo. Datos del plan de corte del archivo variantes de corte para una materia prima-

Descripción de los campos de datos:

Longitud: Longitud de materia prima o pieza.

Cantidad: Cantidad en existencia o que se desea obtener de materia prima o pieza respectivamente.

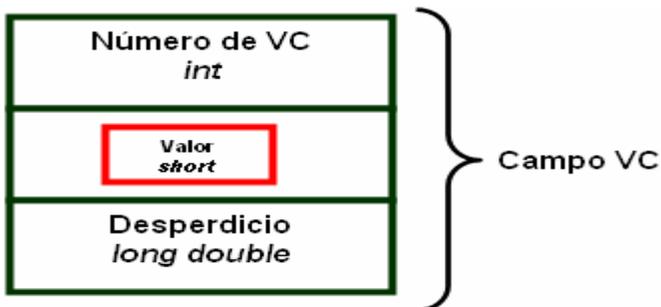


Ilustración 21. -Estructura del archivo. Campo VC del archivo variantes de corte para una materia prima-

Descripción de los campos de datos:

Número de la VC: Número que representa una variante del total.

Valor: Valores de la variante de corte que se interpreta como la cantidad de cada pieza a obtener

Desperdicio: Sobrante de materia prima al aplicar la variante de corte.

El archivo variantes de corte para varias materias prima tiene la misma estructura que el anteriormente analizado sólo se le incluye un campo en el encabezado **Cant_MP : int**.

2.3.3 Diagrama de implementación.

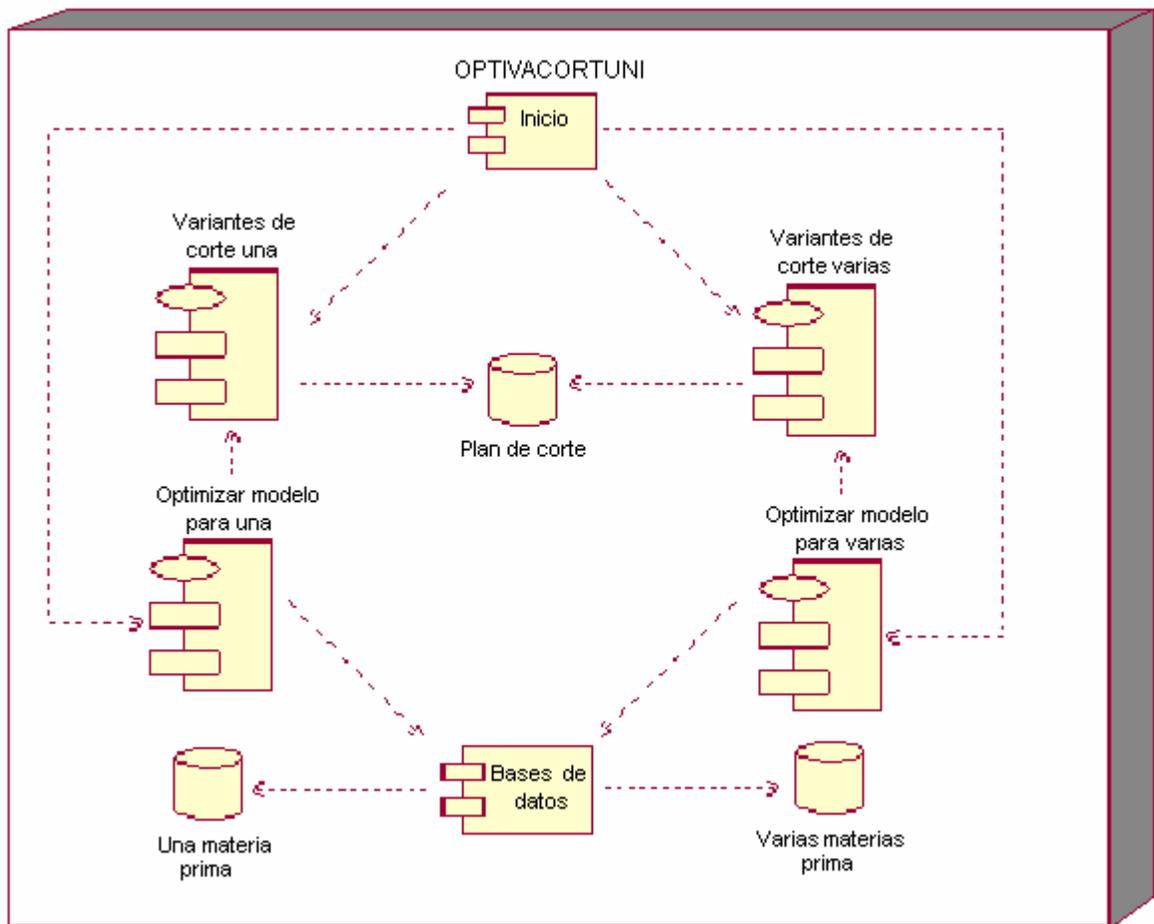


Ilustración 22. -Diagrama de implementación-

2.3.4 Principios de diseño.

La fase de diseño es la que le sigue a la de análisis y la que se antepone a la de implementación. En esta fase se obtiene una solución lógica sustentada en el paradigma de la programación orientada a objetos. “En el diseño modelamos el sistema y encontramos su forma (incluida la arquitectura) para que soporte todos los requisitos- incluyendo los requisitos no funcionales y otras restricciones-que se le suponen”.^[12]

Esta fase de diseño tiene como objetivo llegar a una **mayor claridad** en cuanto a restricciones relacionadas con: requerimientos no funcionales, lenguajes de programación, tecnologías utilizadas, objetos reutilizables y otros, **definir y delimitar las obligaciones** de cada componente, así como las asociaciones entre ellos y

^[12]Jacobson, Ivar. El Proceso Unificado de Desarrollo de Software (Tomo 1).--La Habana: Editorial Félix Varela, 2004.--p.205.

descomponer y distribuir el trabajo en piezas más compactas, con un nivel de abstracción mejor definido para facilitar y agilizar el proceso de implementación.

2.3.4.1 Estándares en la interfaz de la aplicación propuesta.

Un estudio, aplicando la técnica de la entrevista a personas de experiencia relacionadas con el tema, permitió llegar a conclusiones en relación con la interfaz del sistema.

La interfaz que se propone cumple con los estándares actuales del sistema operativo Windows, consistente en: ventanas con barras de menús, botones rápidos, comandos estándares existentes en ese sistema y fácil usabilidad.

Debido al tema que trata la aplicación, se escogieron colores tenues, que no son llamativos y que resultan agradables para la vista de los clientes.

Se tuvo en cuenta la visualización de varias ventanas a la vez que contienen reportes, y además, la visualización de varios reportes a la vez en una misma ventana.

2.3.4.2 Tratamiento de errores.

El sistema comprende e implementa un tratamiento de errores con el fin de que no se acepten valores indebidos introducidos por el usuario.

Para ello se limitan las posibilidades de introducir información errónea por parte del cliente. La entrada referente a los planes de corte se encontrará validada mostrando mensajes de errores cuando se haya cometido alguno.

Los reportes de sólo lectura son el resultado del procesamiento del sistema y se muestran mensajes de información al cliente si se quiere acceder a alguno y no hay datos disponibles para ello.

Si se desean cargar datos desde un archivo y este no es del tipo que procesa la aplicación se mostrará un mensaje de error indicando que el archivo no es del tipo que pueda ser procesado por el sistema.

2.3.4.3 Concepción general de la ayuda.

La ayuda constituye una parte importante del sistema, permitiendo orientar al cliente en las funcionalidades y operaciones que brinda la aplicación.

La búsqueda en la ayuda se hará por contenido o por índice; organizada por módulos y que contenga, además, una explicación general sobre el sistema. Al ser desplegado cada módulo aparecerán todas las opciones que brindan los mismos con sus respectivas explicaciones.

2.4 Conclusiones.

En el presente capítulo, se documentó todo lo referente al desarrollo (descripción y construcción) del sistema propuesto. Para ello se siguió la metodología RUP.

Se hizo un análisis detallado de la estructura y la dinámica del entorno que dio origen a la investigación, consiguiendo una mayor claridad en el problema a resolver. Se introdujo al cliente como parte importante en el proceso de desarrollo obteniendo incrementalmente una mayor información del mismo. Se definieron los componentes básicos que integran el sistema, se analizaron las funcionalidades, los requisitos que debe cumplir y se estableció al usuarios que asumirá los roles fundamentales. Como resultado quedan definidos el modelo del dominio y el modelo del sistema.

Se definieron los conceptos fundamentales en la fase de diseño tales como diagrama de clases del diseño, diagramas del modelo lógico y físico de datos, diagrama de implementación para representar los elementos fundamentales y los principios de diseño utilizados.

Este capítulo resume todo lo referente al proceso de desarrollo del sistema propuesto, encontrándose las dos últimas fases, implementación y prueba, implícitas en la misma génesis del sistema.

Capítulo 3 - Estudio de factibilidad y validación del sistema.

3.1 Introducción.

Este capítulo trata todo lo referente al estudio de la factibilidad del producto. Se estiman el esfuerzo humano y el tiempo de desarrollo que se requieren para la elaboración del mismo, así como los costos y los beneficios tangibles e intangibles que reporta la utilización del sistema. Se hace un análisis entre los costos y los beneficios para concluir si es o no factible el desarrollo del mismo. Estas estimaciones se realizan a través del método de puntos de función del modelo de COCOMO II. Finalmente, se realiza la validación del sistema que se propone a partir de una situación real.

3.2 Estudio de factibilidad.

3.2.1 Planificación por puntos de función.

Para estimar el tamaño del producto y el esfuerzo asociado a su desarrollo se utiliza el método de estimación por puntos de función.

Para realizar el cálculo de los costos de desarrollo del sistema se analizan las cantidades de entradas, salidas, peticiones, archivos lógicos e interfaces externas preliminares que tiene el sistema. También se tuvo en cuenta la conversión al C++, que es de 53 puntos.

Después de este estudio se obtienen los siguientes resultados:

Nombre de la entrada externa	Cantidad de ficheros	Cantidad de Elementos de datos	Clasificación (Bajo, Medio y Alto)
Introducir plan de corte para una longitud de barra de acero en existencia.	1	4	Bajo
Introducir plan de corte para varias longitudes de barras de acero en existencia.	1	5	Bajo
Generar fichero, de las variantes de corte a tener en cuenta en el modelo, referente a una longitud de barra de acero en existencia.	1	5	Bajo
Generar fichero, de las variantes de corte a tener en cuenta en el modelo, referente a varias longitudes de barras de acero en existencia.	1	6	Bajo

Tabla 25. -Planificación. Entradas externas-

Nombre de la salida externa	Cantidad de ficheros	Cantidad de Elementos de datos	Clasificación (Bajo, Medio y Alto)
Emitir reporte de todas las variantes de corte generadas (primarias y de sustitución) para una barra de acero en existencia.	1	4	Bajo
Emitir reporte de todas las variantes de corte primarias generadas para una barra de acero en existencia.	1	4	Bajo
Emitir reporte de todas las variantes de corte de sustitución generadas para una barra de acero en existencia.	1	4	Bajo
Emitir reporte de todas las variantes de corte independientes generadas (primarias y de sustitución) para una barra de acero en existencia.	1	4	Bajo
Emitir reporte de todas las variantes de corte independientes primarias generadas, para una barra de acero en existencia.	1	4	Bajo
Emitir reporte de todas las variantes de corte independientes y de sustitución generadas, para una barra de acero en existencia.	1	4	Bajo
Emitir reporte de todas las variantes de corte dependientes generadas (primarias y de sustitución), para una barra de acero en existencia.	1	4	Bajo
Emitir reporte de todas las variantes de corte dependientes primarias generadas, para una barra de acero en existencia.	1	4	Bajo
Emitir reporte de todas las variantes de corte dependientes y de sustitución generadas, para	1	4	Bajo

una barra de acero en existencia.			
Emitir reporte de eficiencia y variables a contar en el modelo para una barra de acero en existencia.	1	4	Bajo
Emitir reporte de todas las variantes de corte generadas (primarias y de sustitución), para varias barras de acero en existencia.	1	5	Bajo
Emitir reporte de todas las variantes de corte primarias generadas, para varias barras de acero en existencia.	1	5	Bajo
Emitir reporte de todas las variantes de corte de sustitución generadas, para varias barras de acero en existencia.	1	5	Bajo
Emitir reporte de todas las variantes de corte independientes generadas (primarias y de sustitución), para varias barras de acero en existencia.	1	5	Bajo
Emitir reporte de todas las variantes de corte independientes primarias generadas, para varias barras de acero en existencia.	1	5	Bajo
Emitir reporte de todas las variantes de corte independientes y de sustitución generadas, para varias barras de acero en existencia.	1	5	Bajo
Emitir reporte de todas las variantes de corte dependientes generadas (primarias y de sustitución), para varias barras de acero en existencia.	1	5	Bajo
Emitir reporte de todas las variantes de corte dependientes primarias generadas, para varias barras de acero en existencia.	1	5	Bajo
Emitir reporte de todas las variantes de corte dependientes y de sustitución generadas, para	1	5	Bajo

varias barras de acero en existencia.			
Emitir reporte de eficiencia y variables a contar en el modelo para varias barras de acero en existencia.	1	5	Bajo
Emitir reporte del modelo matemático para una barra de acero en existencia.	1	5	Bajo
Emitir reporte del plan de corte optimizado para una barra de acero en existencia.	1	5	Bajo
Emitir reporte práctico del plan de corte optimizado para una barra de acero en existencia	1	5	Bajo
Emitir reporte del modelo matemático para varias barras de acero en existencia.	1	6	Bajo
Emitir reporte del plan de corte optimizado para varias barras de acero en existencia.	1	6	Bajo
Emitir reporte práctico del plan de corte optimizado para varias barras de acero en existencia.	1	6	Bajo

Tabla 26. -Planificación. Salidas externas-

Nombre de la petición	Cantidad de ficheros	Cantidad de Elementos de datos	Clasificación (Bajo, Medio y Alto)
Calcular variantes de corte primarias para una barra de acero en existencia.	1	4	Bajo
Calcular variantes de corte de sustitución para una barra de acero en existencia.	1	4	Bajo
Calcular variantes de corte primarias para varias barras de acero en existencia.	1	5	Bajo
Calcular variantes de corte de sustitución para varias barras de acero en existencia.	1	5	Bajo
Calcular dependencias entre las variantes de corte primarias.	2	9	Medio
Calcular dependencias entre las variantes de corte de sustitución.	2	9	Medio
Suprimir las variantes de corte dependientes (aquellas que resultan de la combinación lineal de otras).	4	20	Alto
Elaborar la forma canónica del modelo matemático.	2	11	Medio
Llevar el modelo matemático de la forma canónica a la estándar.	2	11	Medio
Optimizar la forma estándar del modelo matemático.	2	11	Medio

Tabla 27. -Planificación. Peticiones-

Nombre del fichero interno	Cantidad de records	Cantidad de Elementos de datos	Clasificación (Bajo, Medio y Alto)
Plan de corte para una longitud de materia prima	1	4	Bajo
Plan de corte para varias longitudes de materia prima	1	5	Bajo
Variantes de corte para un longitud de materia prima	1	5	Bajo
Variantes de corte para varias longitudes de materia prima	1	6	Bajo

Tabla 28. -Planificación. Ficheros internos-

Elementos	Bajos	X Peso	Medios	X Peso	Altos	X Peso	Subtotal de puntos de función
Ficheros lógicos internos	4	7	0	10	0	15	28
Ficheros de interfaces externas	0	5	0	7	0	10	0
Entradas externas	4	3	0	4	0	6	12
Salidas externas	26	4	0	5	0	7	104
Peticiones	4	3	5	4	1	6	38
Total							182

Tabla 29. -Planificación: Punto de función-

Características	Valor
Puntos de función desajustados	182
Lenguaje	C++
Instrucciones fuentes por puntos de función	53
Por ciento de la aplicación en cuanto a requerimientos funcionales	100%
Total de instrucciones fuentes	9646

Tabla 30. -Planificación: Miles de instrucciones fuentes-

3.2.2 Determinación de los costos.

Cálculo del esfuerzo, tiempo de desarrollo, cantidad de hombres y costo.

Cálculo de:	Valor	Justificación
RCPX	1 (Nominal)	No se requiere de amplia documentación. El sistema tiene una moderada complejidad.
RUSE	1,07 (Alto)	Se implementa código reusable para el aprovechamiento de este en toda la aplicación.
PDIF	1 (Nominal)	No tiene grandes restricciones en cuanto al tiempo de ejecución ya que el sistema podrá estar trabajando varias horas. El sistema tiene limitación de memoria. La plataforma de aplicación tiene gran estabilidad.
PERS	0,83 (Alto)	Hay poco movimiento del personal.
PREX	1 (Alto)	El equipo tiene buen dominio y posee conocimiento del lenguaje de programación, con una experiencia de aproximadamente cuatro años.
FCIL	0,87 (Alto)	Se utilizan herramientas como: Borland Builder C++, así como CASE Rational Rose para la documentación, empleando como notación UML.
SCED	1 (Nominal)	La planificación se hace con moderada frecuencia.
PREC	2,48	El equipo de desarrollo posee una comprensión considerable

	(Alto)	de los objetivos del producto.
FLEX	3,04 (Nominal)	El sistema cuenta con alguna flexibilidad en relación con las especificaciones de los requerimientos preestablecidos y a las especificaciones de interfaz externa.
TEAM	2,19 (Alto)	El equipo que va a desarrollar el sistema es altamente cooperativo.
RESL	4,24 (Nominal)	Teniendo en cuenta la alta experiencia que existe en el país acerca de este tipo de estudios existen algunos factores de riesgo.
PMAT	3,12 (Alto)	Se encuentra en su primera etapa un poco avanzada.

Tabla 31. -Costos: Factores de escalas-

Multiplicador de esfuerzos

$$EM = \prod_{i=1}^7 E_{mi} = RCPX * RUSE * PDIF * PERS * PREX * FCIL * SCED$$

$$EM = \prod_{i=1}^7 E_{mi} = 1 * 1,07 * 1 * 0,83 * 1 * 0,87 * 1 = 0,7726 \approx 0,77$$

Factores de escala

$$SF = \sum SFi = PREC + FLEX + RESL + TEAM + PMAT$$

$$SF = \sum SFi = 2,48 + 3,04 + 2,19 + 4,24 + 3,12 = 15,07$$

Valores de los coeficientes

$$A = 2,94$$

$$B = 0,91$$

$$C = 3,67$$

$$D = 0,24$$

$$E = B + 0,01 * SF = 0,91 + 0,01 * 15,07 = 1,06$$

$$F = D + 0,2 * (E - B) = 0,24 + 0,2 * (1,06 - 0,91) = 0,27$$

Esfuerzo

$$PM = A * (MF)^E * EM = 2,94 * (8,538)^{1,06} * 0,77 = 21,98$$

Cálculo del tiempo de desarrollo

$$TDEV = C * PM^F = 3,67 * (21,98)^{0,27} = 8,45$$

Cálculo de la cantidad de hombres

$$CH = PM / TDEV = 21,98 / 8,45 = 2,6$$

Recalculando

$CH = PM / TDEV$
 $TDEV = PM/CH$
 $CH = 2$
 $TDEV = 21,98/1=21,98 \approx 22$ meses

Costo

Se asume como salario promedio mensual 225\$
 $CHM = 1 * \text{Salario Promedio} = 1 * 225 = 225$ \$/mes
 $\text{Costo} = CHM * PM = 225 * 21,98 = 4945$

Los costos en los que se incurriría, de desarrollarse el sistema, son:

Cálculo de:	Valor
Esfuerzo(PM)	21,98
Tiempo de desarrollo	22 meses
Cantidad de hombres	1
Costo	4945
Salario medio	225
RCPX	1
RUSE	1,07
PDIF	1
PREX	2,48
FCIL	0,87
SCED	1

Tabla 32. -Costos totales-

3.2.3 Beneficios tangibles e intangibles.

Todo proyecto genera efectos económicos. Estos efectos se clasifican en: beneficios tangibles y beneficios intangibles, entre otros. Los proyectos informáticos también producen estos efectos.

Los beneficios intangibles son aquellos que resultan difíciles o prácticamente imposibles de ponderar en unidades monetarias.

Los beneficios tangibles que pueden producir los proyectos informáticos son:

- Ahorro de horas-hombre (por no tener que contratar personal adicional o por el aumento de la productividad).
- Venta de información (venta de software).
- Ahorro en arriendo de oficinas (dado el paso a medios magnéticos de archivo a carpetas que existen físicamente).
- Ahorro en costos de operación (dejar de pagar servicios a empresas, disminución de costos de mantenimiento).
- Valor residual de los equipos.

- Ahorro de horas-hombre del personal que actualmente labora en el sistema (aumento de la productividad).
- Ahorro de horas-hombre de los clientes (beneficio social).

Los beneficios obtenidos con el desarrollo de este sistema son fundamentalmente intangibles, ya que este permite orientar cómo se debe llevar a cabo el corte de materiales garantizando el desperdicio óptimo, para así optimizar el proceso de corte de barras de acero en el MINCONS. Todo esto se lleva a cabo en un ambiente sencillo, moderno y actualizado. Además, el sistema es compatible con el sistema operativo Windows y presenta algoritmos optimizados para que pueda ser utilizado por cualquier usuario, en especial, por el jefe de producción del MICONNS.

El sistema contempla la opción de aceptar planes de corte que consideren la existencia de un solo tipo de longitud de barra de acero y diferentes tipos de longitudes de barras de acero, de manera que se puedan comparar (si se estima conveniente) los resultados arrojados para uno y para otros, con el objetivo de tomar decisiones que puedan influir significativamente en el proceso de corte.

3.2.4 Análisis de costos y beneficios.

Este sistema, como resultado del presente trabajo de diploma, no implica costo alguno para el MICONNS, sin embargo, al desarrollo de todo producto informático va asociado un costo y el justificarlo depende de los beneficios tangibles e intangibles que produce.

La utilización de este nuevo sistema permite orientar cómo se debe realizar el corte de materiales para garantizar el desperdicio óptimo. De esta forma permite optimizar el proceso de corte de barras de acero en el MINCONS.

Además, posibilita aprovechar las potencialidades informáticas existentes en la empresa, en función del perfeccionamiento del proceso de producción.

Para la realización de este sistema no fue necesaria una inversión en los medios técnicos.

3.3 Validación.

En este epígrafe se presenta el análisis realizado, a partir del sistema informático propuesto, en el proceso de corte de barras de acero de diámetro 12 milímetros para la

construcción de piezas prefabricadas en una entidad económica perteneciente al MICONS en la ciudad de Cienfuegos.

La tarea de la entidad económica que se asume a partir del software es la siguiente: a partir de barras de acero de longitud de 9 metros se deben cortar piezas de longitud 3.43, 3.36, 2.40, 2.34, 1.79 y 1.59 metros y se necesitan 126, 95, 245, 126, 310 y 442 piezas respectivamente.

1. Sección de la correspondiente entrada de datos. Sólo es preciso contar con elementales conocimientos informáticos.

The screenshot shows a software window titled "Entrada de Datos (Una Materia Prima)". The window has a blue title bar with a close button. Below the title bar, the text "Unidad de Medida" is displayed. The main area is divided into two sections: "Materia Prima" and "Piezas a Cortar".

Materia Prima

Entre la Longitud de la Materia Prima:

Entre la Cantidad de Materia Prima:

Piezas a Cortar

Entre la Longitud de la Pieza:

Necesidad de la Pieza:

Materia Prima: 9 m

Pieza#1: 3,43 m - (126)
Pieza#2: 3,36 m - (95)
Pieza#3: 2,4 m - (245)
Pieza#4: 2,34 m - (126)
Pieza#5: 1,79 m - (310)
Pieza#6: 1,59 m - (442)

Ilustración 23. -Entrada de datos para una situación real-

- Sección correspondiente a la modelación matemática de la situación en estudio aclarando que el cliente no necesita ver esto, pero está a su alcance.

MODELO MATEMÁTICO

MIN DMP = 0,35 X1 + 0,77 X2 + 0,05 X3 + 0,20 X4 + 0,06 X5 + 0,01 X6 + 0,04 X7 + 0,05 X8 + 0,07 X1,9 + 0,05 X1,10 + 0,96 X2,11 +

SUJETO A:

2 X1 + 1 X2 + 1 X3 + 1 X4 + 0 X5 + 0 X6 + 0 X7 + 0 X8 + -1 X1,9 + -1 X1,10 + 0 X2,11 + 0 X2,12 + 0 X3,13 + 0 X4,14 + 0 X5,15 + 0 X1 + 0 X2 + 0 X3 + 0 X4 + 1 X5 + 0 X6 + 0 X7 + 0 X8 + 1 X1,9 + 0 X1,10 + -1 X2,11 + -1 X2,12 + 0 X3,13 + 0 X4,14 + 0 X5,15 + 0 X1 + 2 X2 + 0 X3 + 0 X4 + 1 X5 + 3 X6 + 1 X7 + 0 X8 + 0 X1,9 + 0 X1,10 + 1 X2,11 + 0 X2,12 + -1 X3,13 + 0 X4,14 + 0 X5,15 + 0 X1 + 0 X2 + 1 X3 + 0 X4 + 0 X5 + 0 X6 + 0 X7 + 0 X8 + 0 X1,9 + 0 X1,10 + 0 X2,11 + 0 X2,12 + 1 X3,13 + -1 X4,14 + 0 X5,15 + 1 X1 + 0 X2 + 0 X3 + 3 X4 + 0 X5 + 1 X6 + 1 X7 + 5 X8 + 0 X1,9 + 1 X1,10 + 0 X2,11 + 0 X2,12 + 0 X3,13 + 1 X4,14 + -1 X5,15 + 0 X1 + 0 X2 + 2 X3 + 0 X4 + 2 X5 + 0 X6 + 3 X7 + 0 X8 + 0 X1,9 + 1 X1,10 + 0 X2,11 + 2 X2,12 + 0 X3,13 + 0 X4,14 + 1 X5,15

$X_i \geq 0$
 $X_{k,i} \geq 0$

DONDE:

DMP: Desperdicio total de materia prima.
 X_i : Cantidad de materia prima de longitud 9 m que es necesario cortar, según el patrón de corte i.
 $X_{k,i}$: Cantidad de pieza número k que es necesario cortar, según el patrón de corte i.
 $i = 1, \dots, 16$
 $k = 1, \dots, 6$

VARIABLE	VARIANTE DE CORTE
X1	1
X2	5
X3	11
X4	12
X5	21
X6	29
X7	43
X8	54
X1,9	60

Ilustración 24. -Modelación matemático para una situación real-

3. Sección con el plan de corte que recibe el cliente. Este reporte, expresado en un lenguaje en total correspondencia con el utilizado en el plan de corte a ejecutar, es emitido por el software y es el principal resultado para la tarea que se debe orientar a la persona que realmente lleva a cabo el corte de acero. Este reporte final (se nombra reporte práctico) tiene implícito (a partir de los datos iniciales que fueron tecleados y presentados en la primera sección) la modelación matemática del problema, el procesamiento computacional del modelo matemático y la interpretación económica de la solución óptima.



Ilustración 25. -Reporte práctico para una situación real-

3.4 Conclusiones.

El sistema propuesto trae consigo una serie de beneficios, sobre todo intangibles, para entidades pertenecientes al MICONS. Este va a contribuir positivamente en el funcionamiento del proceso de producción, específicamente en el proceso de corte de barras de acero, lo que indica que es factible implementar el sistema propuesto.

Una vez terminado el estudio de factibilidad de este sistema, se estima un tiempo de 22 meses para su construcción por un hombre y su costo asciende a \$4945.

La validación del software, a partir de una situación real en determinada entidad económica de la provincia de Cienfuegos perteneciente al MICONS avala la utilización práctica del mismo.

Conclusiones.

1. La utilización del sistema informático “OPTIVACORTUNI” (que integra la modelación matemática del problema, el procesamiento computacional del modelo matemático y la interpretación económica de la solución óptima) es una alternativa factible (a partir de elementales conocimientos informáticos) para ser utilizado en la optimización del proceso de corte de barras de acero en entidades pertenecientes al MICONS.
2. La metodología RUP como guía a utilizar en el proceso de desarrollo, la utilización del lenguaje de programación C/C++ para la implementación y la herramienta informática Borland C++ Builder 6.0 pueden permitir el desarrollo de software con características análogas (en cuanto a metodología de desarrollo, restricciones de memoria y capacidad de procesamiento y ambiente visual al estilo Windows) a las del sistema informático desarrollado.
3. El informe de este trabajo de diploma se puede convertir en un documento de consulta para la dirección de empresas que en el desarrollo del proceso productivo estén relacionadas con el corte unidimensional de materiales.

Recomendaciones.

1. Utilizar el sistema informático “OPTIVACORTUNI” (que integra la modelación matemática del problema, el procesamiento computacional del modelo matemático y la interpretación económica de la solución óptima) en la optimización del proceso de corte de barras de acero en entidades pertenecientes al MICONS.
2. Analizar la posibilidad de utilizar el sistema informático desarrollado en otras entidades económicas que en su quehacer productivo estén relacionadas con el corte unidimensional de materiales.
3. Considerar el código fuente del sistema informático “OPTIVACORTUNI” como elemento de referencia al desarrollar software (matemáticos) con características análogas.
4. Utilizar el sistema informático en el desarrollo del Proceso Docente Educativo (PDE) de la disciplina *Matemática Aplicada* de la carrera de Ingeniería Informática. Extenderlo, también a otras carreras que tienen incluido en el programa de estudio la Investigación de Operaciones (en particular el tema Programación en Enteros).
5. Valorar (a partir de necesidades prácticas que puedan aparecer en la vida empresarial) la posibilidad de incluir en el sistema informático desarrollado otros tipos de restricciones y/o consideraciones para la función objetivo.
6. Utilizar el informe de este trabajo investigativo como documento de consulta en la empresa (en su quehacer productivo) y desarrollar seminarios con los directivos que están directamente relacionados con la actividad en caso que se considere necesario.

Bibliografía.

Álgebra Lineal/María Virginia Varela Marcelo...[et.al.]-- La Habana: Editorial Félix Varela, 2003.--487p.

Arencibia Rodríguez del Rey, Yailem. OrtoGram: Sistema informático para el trabajo de la ortografía en la enseñanza primaria/Yailem Arencibia Rodríguez del Rey, Annia Pérez Fernández; Viviana Toledo, tutor.--Trabajo de Diploma: UCF (Cf), 2007.--106h. : illus.

Branch and Bound. Enciclopedia Wikipedia. Tomado de: http://es.wikipedia.org/wiki/branch_and_bound, 10 de abril 2008.

Budd, Timothy A. Introducción a la Programación Orientada a Objetos/Timothy A. Budd.--La Habana: Editorial de Libros Para la Educación, 2005.--409p.

Ciencia. Tomado de: Diccionario Encarta @ 2006, 20 de mayo 2008.

de León Rodríguez, Narciso Rubén. Métodos Matemáticos en la Dirección de los Procesos de Corte de Materiales/Narciso Rubén de León Rodríguez.--Tesis en Opción al Grado Científico de Master en Matemática Aplicada: UCF (Cf), 1996.--67h. : illus.

Dietel, Harvey M. Como Programar en C-C++/Harvey M. Dietel, Paul J. Dietel.--La Habana: Editorial de Libros Para la Educación, ..[199?].--3t.

Hernández Sampier, Roberto. Metodología de la Investigación 1/Roberto Hernández Sampier.--La Habana: Editorial Félix Varela, 2004.--475p.

Heurísticas. Enciclopedia Wikipedia. Tomado de: <http://es.wikipedia.org/wiki/Heur%C3%ADstica>, 15 de enero 2008.

Introducción a la investigación de operaciones.--Ciudad de La Habana: ..[s.n.], ..[200?].-3t.

Investigación de Operaciones. Enciclopedia Wikipedia. Tomado de: <http://es.wikipedia.org/IO>, 12 de mayo 2008.

Jacobson, Ivar. El Proceso Unificado de Desarrollo de Software/Ivar Jacobson, Grady Booch, James Rumbaugh.--La Habana: Editorial Félix Varela, 2004.--2t.

Java (informática). Tomado de: Enciclopedia Encarta @ 2006, 3 de mayo de 2008.

Larman, Graig. UML y Patrones: Introducción al análisis y diseño orientado a objetos/Graig Larman.--La Habana: Editorial Félix Varela, 2004.--2t.

Lenguaje de programación C. Enciclopedia Wikipedia. Tomado de: http://es.wikipedia.org/wiki/Lenguaje_C, 27 de enero 2008.

Matemática. Enciclopedia Wikipedia. Tomado de: <http://es.wikipedia.org/wiki/Matem%C3%A1ticas>, 20 de mayo 2008.

Matemática: Décimo grado/Luis Campistrous Pérez...[et.al.]--Ciudad de La Habana: Editorial Pueblo y Educación, 1990.--347p.

Moya Herrera, Susel de Lourdes. Alternativa para el empleo del Cuaderno de Trabajo de Ortografía de 6. grado/Susel de Lourdes Moya Herrera; Débora Elena Rodríguez Gómez, tutor.--Trabajo de Diploma: ISP (Cf), 2005.--56h. : ilus.

Pascal (informática). Tomado de: Enciclopedia Encarta @ 2006, 3 de mayo de 2008.

Programación Matemática I/Pilar Felipe...[et.al.]--Ciudad de La Habana: Empresa Nacional de Producción del Ministerio de Educación Superior, 1983.--351p.

Programación orientada a objetos. Enciclopedia Wikipedia. Tomado de: http://es.wikipedia.org/wiki/Programaci%C3%B3n_orientada_a_objetos, 10 de mayo 2008.

Revolución Industrial. Tomado de: Enciclopedia Encarta @ 2006, 15 de mayo 2008.

Tecnología y Sociedad.-- La Habana: Editorial Félix Varela, 1999.--414p.

Zukowski, John. Programación Java 2 J2SE 1.4/John Zukowski.--La Habana: Editorial Félix Varela, 2006.--3t.

Anexos.

Anexo 1. Algoritmo Branch and Bound.

Algoritmo *B&B* para el problema PLE

0) **Inicialización.**

- Situar PLE en la raíz de \mathcal{T} .
- Incumbente $z_0 \leftarrow \infty$.

1) **Mientras** ($\mathcal{T} \neq \emptyset$):

- Seleccionar un $(\text{PLE})_\ell$ no marcado.
Sea G su conjunto factible.
- Resolver su relajación $(\text{RL})_\ell$.
Sea $x_{RL\ell}^*$ una solución de ésta.

Si ($(\text{RL})_\ell$ infactible ó $[z_{RL\ell}^*] \geq z_0$) **entonces** marcar la hoja ℓ .

Si ($x_{RL\ell}^* \in Z^n$ & $(\text{PLE})_\ell$ no marcado) **entonces**

Si ($z_{RL\ell}^* < z_0$) **entonces**

$x_{PLE}^* \leftarrow x_{RL\ell}^*$ (candidato al óptimo)

$z_0 \leftarrow z_{RL\ell}^*$

Fin si

- Marcar $(\text{PLE})_\ell$

Fin si

Si ($(\text{PLE})_\ell$ no marcado) **entonces**

- Tomar $(x_{RL\ell}^*)_j = \omega \notin Z$.

- Efectuar una partición de $G = G' \cup G''$

(Añadir la restricción $x_j \leq \lfloor \omega \rfloor$ para formar un nodo sucesor del ℓ y la $x_j \geq \lceil \omega \rceil$ para formar el otro nodo sucesor.)

Fin si

Fin Mientras

3) Solución: x_{PLE}^*

Figura 1. -Algoritmo del Branch and Bound-

Anexo 2. Algoritmo Branch and Bound. Desarrollo por niveles.

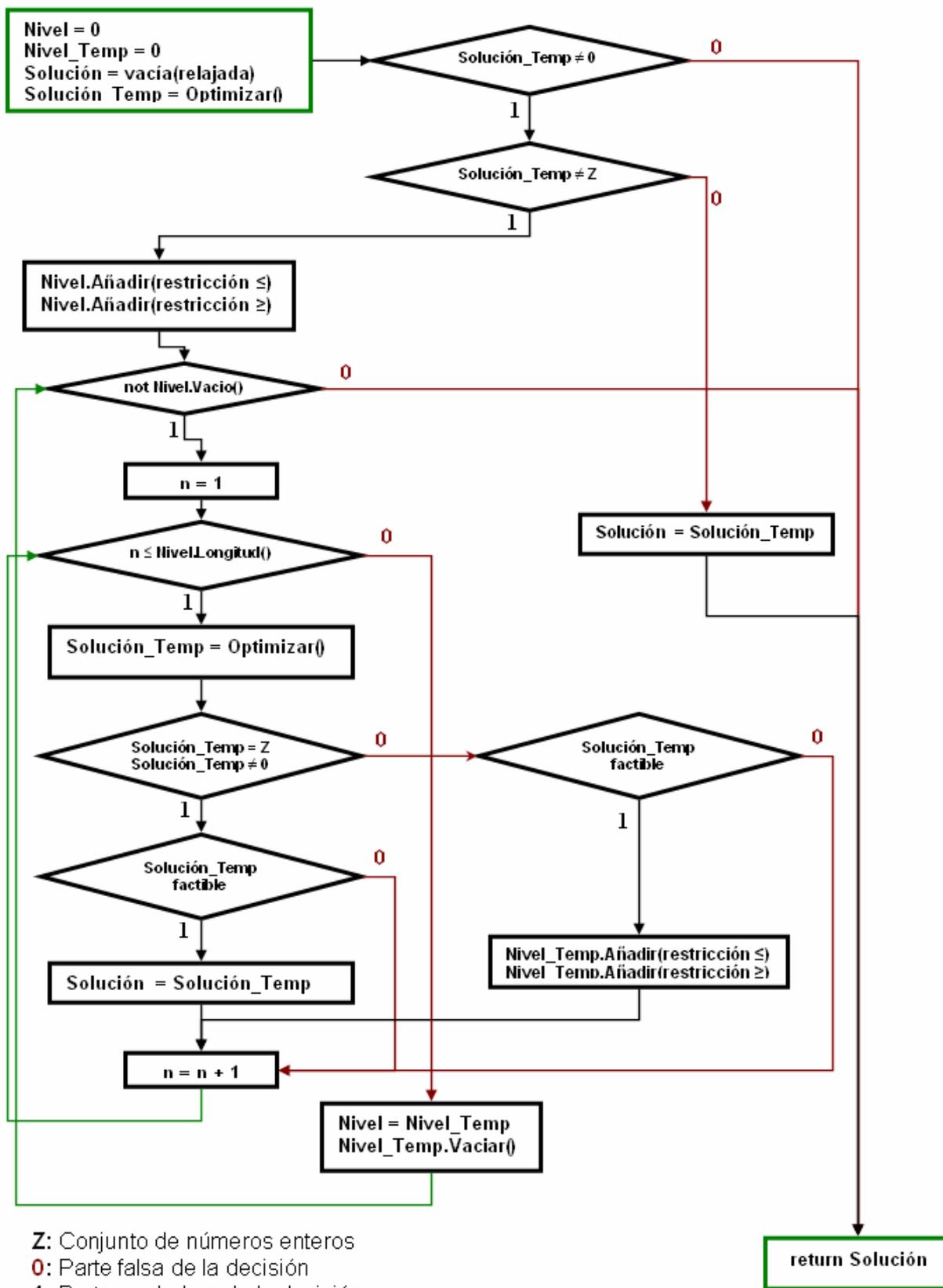


Figura 2. -Algoritmo del Branch and Bound. Desarrollo por niveles-

Anexo 3. Implementación del algoritmo Branch and Bound. Desarrollo por niveles.

```

175 //-----
176 int TBranch_Bound::calcular_solucion()
177 {
178     int no_acotada, pos_var_no_entera;
179     bool tiene_sol;
180
181     //*****
182     //*****
183     inicializar_B();          //calcular solucion
184     no_acotada = optimizar();
185
186     pos_var_no_entera = solucion_entera();
187     tiene_sol = tiene_solucion();
188
189     if(no_acotada===-1 || !tiene_sol)
190         return -1; //no solucion.
191
192     if(pos_var_no_entera==0) //felicidades tu primera solucion fue la solucion entera
193     {
194         no_solucion = false;
195         asignar_solucion();
196         z = valor_de_z();
197     }
198     else
199     {
200         int length_nivel=2, n, k=1;
201         real_largo temp_z, xbi;
202
203         SRestricc R;
204         Nodo_Branch N; //lista de R
205         TListaDobleEnlazada<Nodo_Branch> *Temp_Nivel;
206         Nivel = new TListaDobleEnlazada<Nodo_Branch>();
207
208         xbi = obtener_xbi(pos_var_no_entera);
209         R.asignar( base[pos_var_no_entera-1], menor_igual, floorl(xbi) );
210         N.Añadir(R);
211         Nivel->Añadir(N); //un nuevo problema
212
213
214         N[k].asignar( base[pos_var_no_entera-1], mayor_igual, ceill(xbi) );
215         Nivel->Añadir(N);
216
217         while(length_nivel>0) //si hay posibles nodos que explorar
218         {
219             Temp_Nivel = new TListaDobleEnlazada<Nodo_Branch>();
220             asignar_cant_restric( m_inicial+Nivel->Obtener(1).Longitud() ); //dimensionando el arreglo
221
222             for(n=1; n<=length_nivel; ++n)
223             {
224                 asignar_retricc(Nivel->Obtener(n));
225
226                 inicializar_B(); //calcular solucion
227                 no_acotada = optimizar();
228
229                 pos_var_no_entera = solucion_entera();
230                 tiene_sol = tiene_solucion();
231

```

Figura 3. -Implementación del algoritmo Branch and Bound. Desarrollo por niveles (parte 1)-

```

232         if(pos_var_no_entera==0 && tiene_sol) //posible solucion entera factible
233         {
234             z = valor_de_z();
235             if(no_solucion) //si no hay solucion factible todavia
236             {
237                 asignar_solucion();
238                 Flista_sol_restricc = Nivel->Obtener(n);
239                 no_solucion = false;
240             }
241             else //ya se han encontrado otras soluciones factibles
242             {
243                 temp_z = valor_de_z();
244                 if( temp_z<z)//si es una mejor solucion
245                 {
246                     asignar_solucion();
247                     z = temp_z;
248                     Flista_sol_restricc = Nivel->Obtener(n);
249                 }
250             }
251         }
252         else
253         if(no_acotada==0 && tiene_sol) //si es una solucion no entera pero factible
254         {
255             temp_z = valor_de_z();
256             if( no_solucion || //si no hay solucion todavia, generar dos nuevos problemas
257                (!no_solucion && temp_z<z) //o si hay solucion pero se puede obtener una z mejor
258             )
259             {
260                 xbi = obtener_xbi(pos_var_no_entera);
261                 R.asignar( base[pos_var_no_entera-1], menor_igual, floorl(xbi) );
262                 Nivel->Obtener(n).Anadir(R);
263                 Temp_Nivel->Anadir( Nivel->Obtener(n) );
264
265                 k=Nivel->Obtener(n).Longitud();
266                 Nivel->Obtener(n).Obtener(k).asignar( base[pos_var_no_entera-1],
267                                                         mayor_igual,
268                                                         ceil(l(xbi) ) );
269                 Temp_Nivel->Anadir( Nivel->Obtener(n) );
270             }
271         }
272     }
273
274     delete Nivel;
275     Nivel = Temp_Nivel;
276     length_nivel = Nivel->Longitud();
277 }
278
279 }
280
281 if(Nivel)
282 {
283     delete Nivel;
284     Nivel = NULL;
285 }
286 asignar_cant_restric(m_inicial);
287 Vaciar();
288 solucion.OrdenarBurbuja( Demonios::Comparar_Solve_Var );
289
290 if(no_solucion)
291     return -1; //no se encontro ninguna solucion
292
293 return 0;
294 }
295 //-----

```

Figura 4. -Implementación del algoritmo Branch and Bound. Desarrollo por niveles (parte 2)-

Anexo 4. Prototipos.

Entrada de Datos (Una Materia Prima)

Unidad de Medida

Materia Prima

Entre la Longitud de la Materia Prima:

Entre la Cantidad de Materia Prima:

Piezas a Cortar

Entre la Longitud de la Pieza:

Necesidad de la Pieza:

Materia Prima: 12 m - (300)

Pieza#1: 3,43 m - (20)
Pieza#2: 3,36 m - (35)
Pieza#3: 2,4 m - (30)
Pieza#4: 2,34 m - (47)
Pieza#5: 1,79 m - (30)
Pieza#6: 1,59 m - (15)

Figura 5. -Caso de uso: Introducir plan de corte para una materia prima en existencia-

Entrada de Datos (Varias Materias Primas)

Unidad de Medida

Materias Prima

Longitud de Materia Prima:

Cantidad de Materia Prima:

Mat. Prima#1: 9 m - (100)
Mat. Prima#2: 10,5 m - (80)
Mat. Prima#3: 12 m

Piezas a Cortar

Longitud de Pieza:

Necesidad de la Pieza:

Pieza#1: 3,43 m - (20)
Pieza#2: 3,36 m - (35)
Pieza#3: 2,34 m - (47)
Pieza#4: 1,7 m - (34)

Figura 6. -Caso de uso: Introducir plan de corte para varias materias prima en existencia-

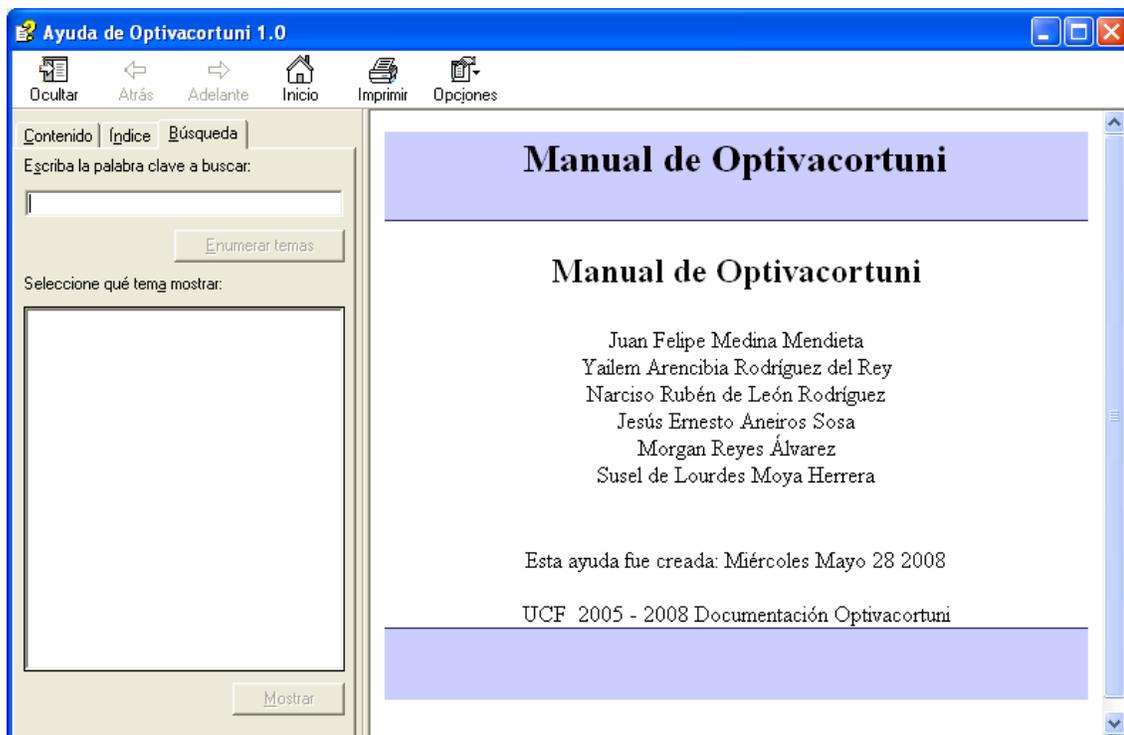


Figura 7. -Caso de uso: Consultar ayuda-

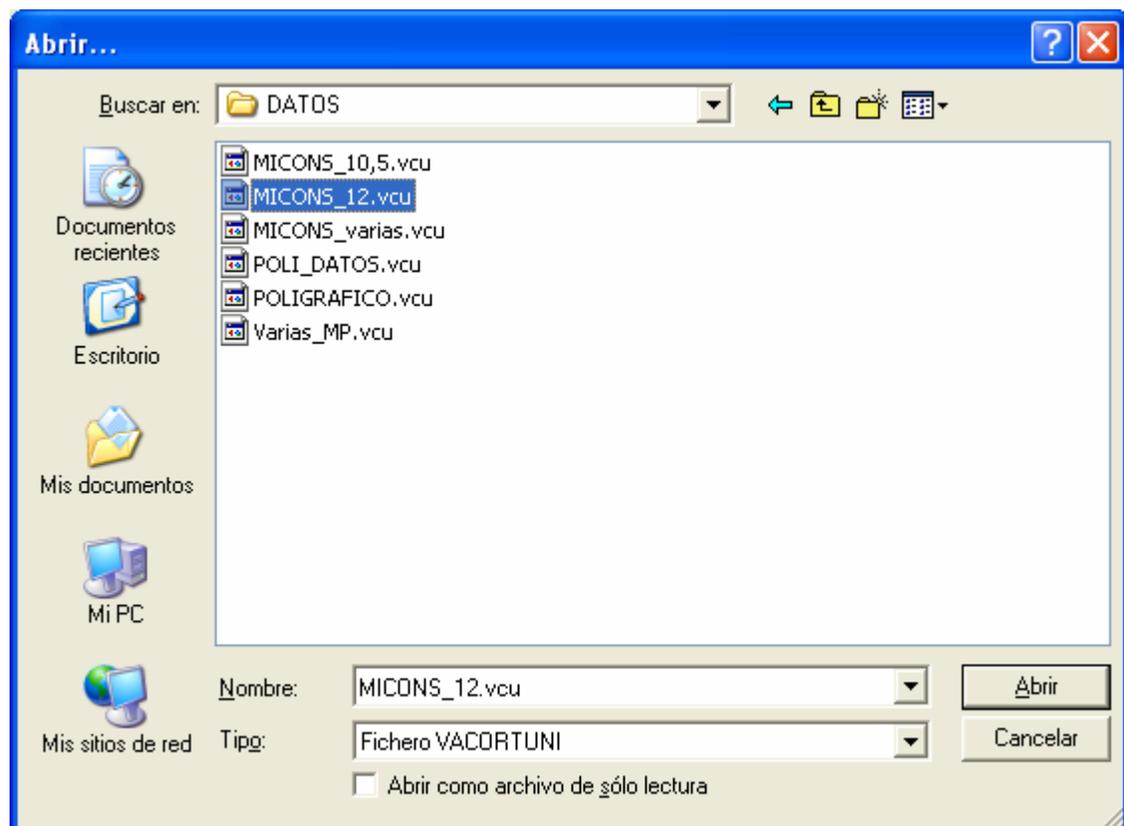


Figura 8. -Caso de uso: Guardar plan de corte-

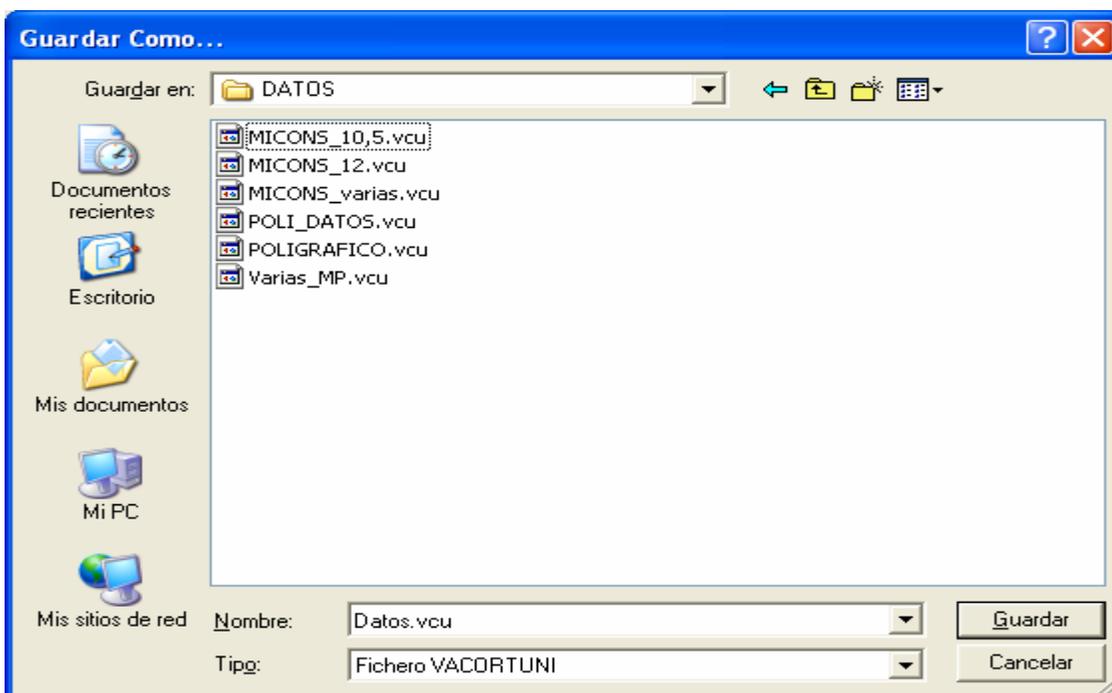


Figura 9. -Caso de uso: Cargar plan de corte-

Reporte(Una Materia Prima)

Variantes de Corte (165)

Longitud de Materia Prima 12,00 m

Pieza#1 3,43 m
 Pieza#2 3,36 m
 Pieza#3 2,40 m
 Pieza#4 2,34 m
 Pieza#5 1,79 m
 Pieza#6 1,59 m

Se generaron 165 variantes de corte

No.VC	Pieza#1	Pieza#2	Pieza#3	Pieza#4	Pieza#5	Pieza#6	Desperdicio(m)
1	3	0	0	0	0	1	0,12
2	2	1	0	0	0	1	0,19
3	2	0	2	0	0	0	0,34
4	2	0	1	1	0	0	0,40
5	2	0	1	0	1	0	0,95
6	2	0	1	0	0	1	1,15
7	2	0	0	2	0	0	0,46
8	2	0	0	1	1	0	1,01
9	2	0	0	1	0	1	1,21
10	2	0	0	0	2	0	1,56
11	2	0	0	0	1	2	0,17
12	2	0	0	0	0	3	0,37
13	1	2	0	0	1	0	0,06
14	1	2	0	0	0	1	0,26
15	1	1	2	0	0	0	0,41
16	1	1	1	1	0	0	0,47

Figura 10. -Caso de uso: Calcular variantes de corte para una materia prima en existencia-

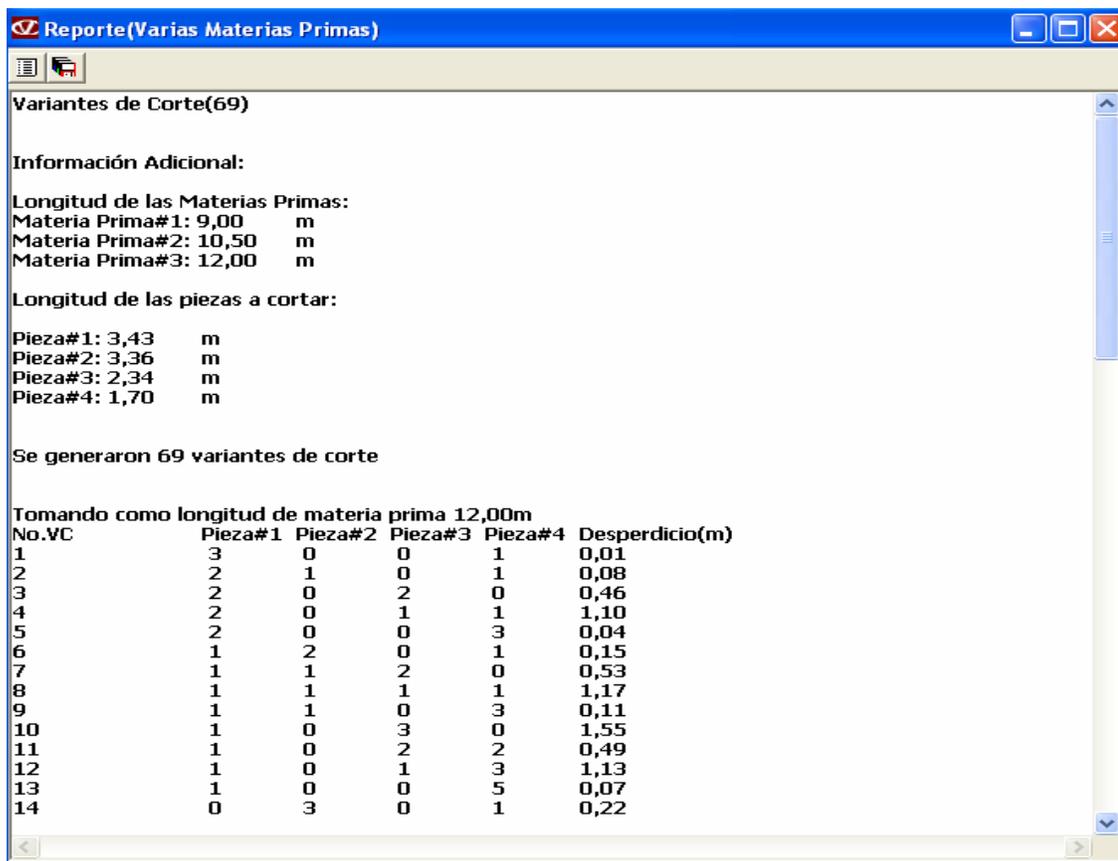


Figura 11. -Caso de uso: Calcular variantes de corte para varias materias prima en existencia-

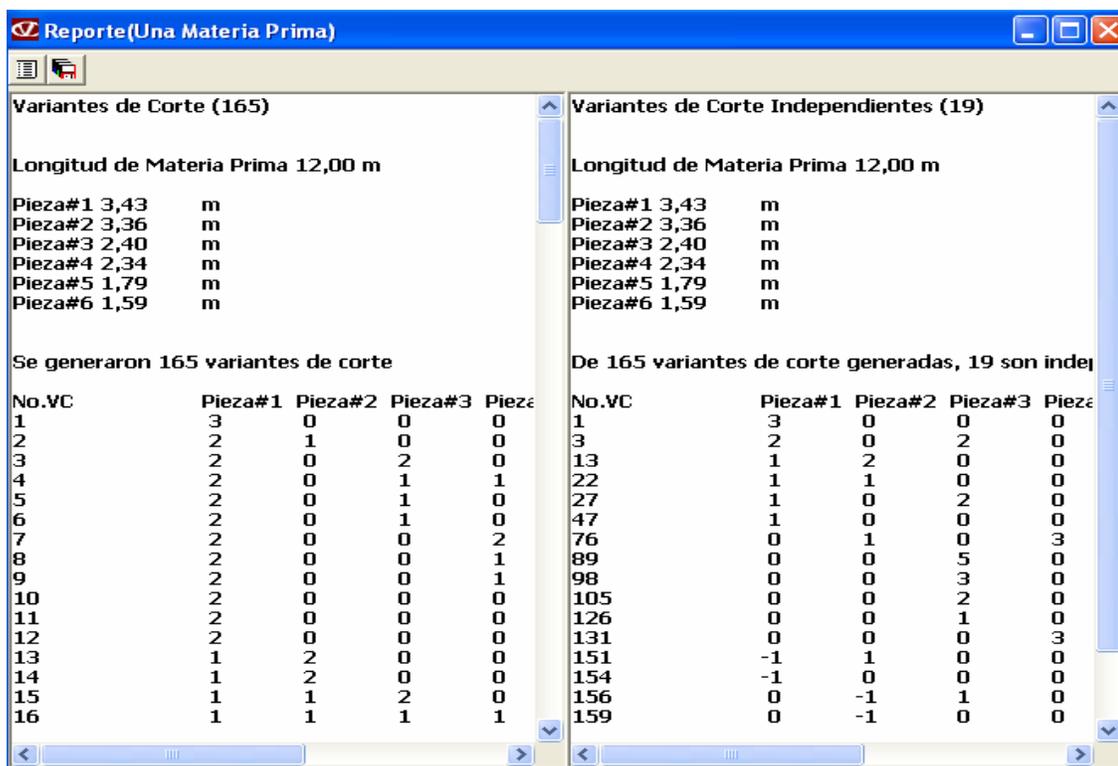


Figura 12. -Caso de uso: Ver reporte referente a las variantes de corte para una materia prima en existencia-

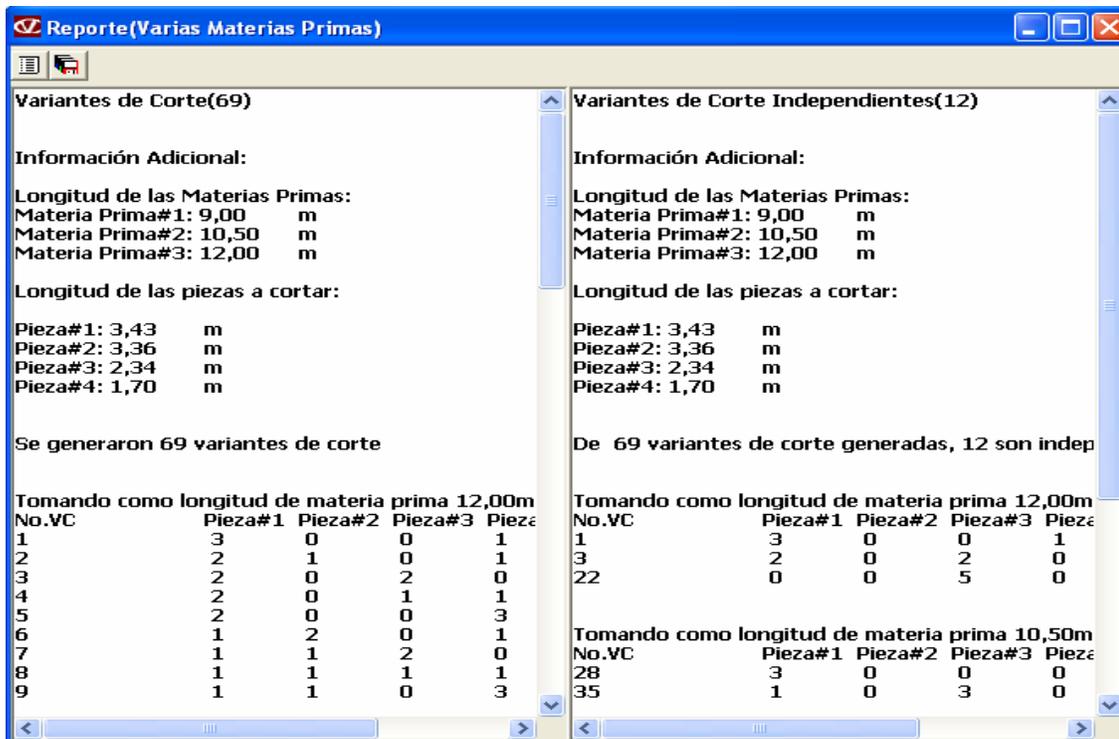


Figura 13. -Caso de uso: Ver reporte referente a las variantes de corte para varias materias prima en existencia-

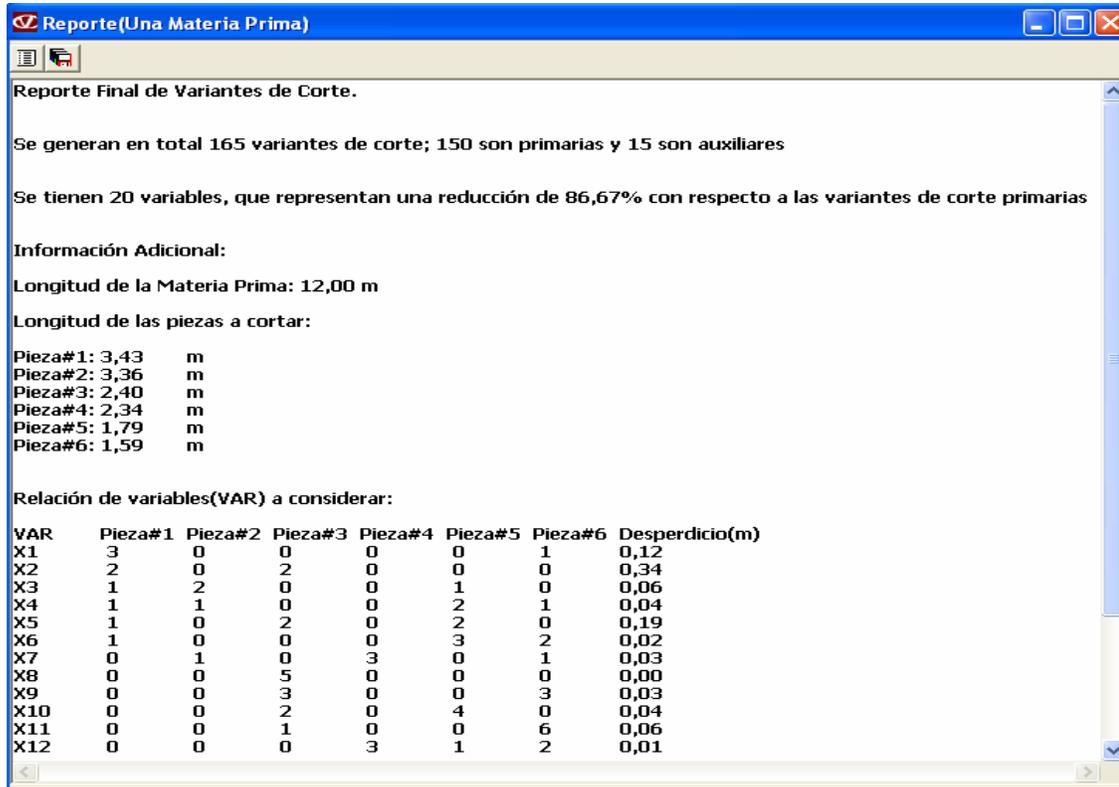


Figura 14. -Caso de uso: Ver reporte de efectividad para una materia prima en existencia-

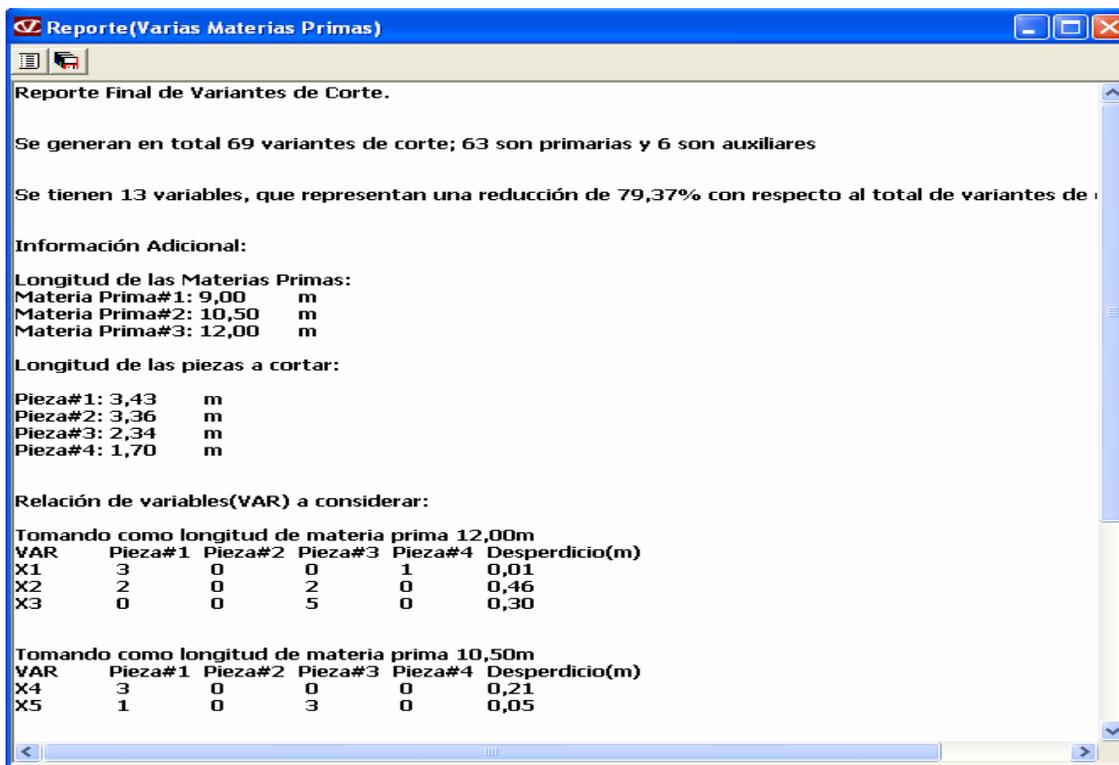


Figura 15. -Caso de uso: Ver reporte de efectividad para varias materias prima en existencia-

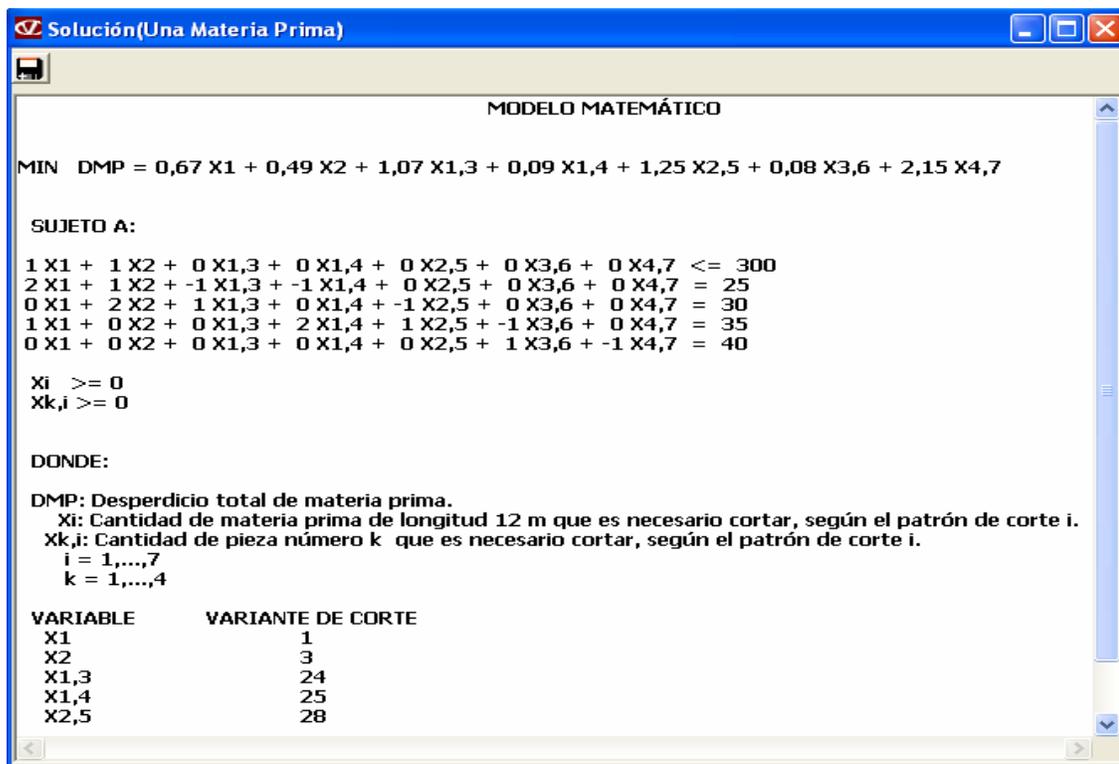


Figura 16. -Caso de uso: Reporte de la forma canónica del modelo matemático para una materia prima en existencia-

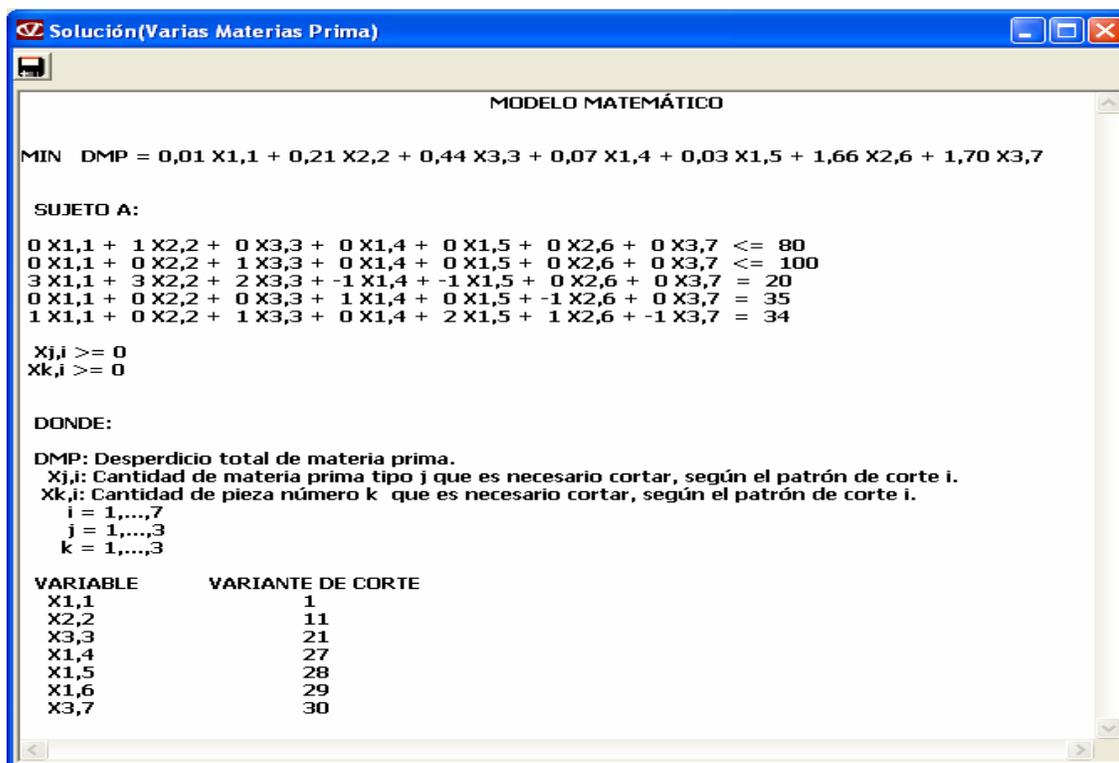


Figura 17. -Caso de uso: Reporte de la forma canónica del modelo matemático para varias materias prima en existencia-

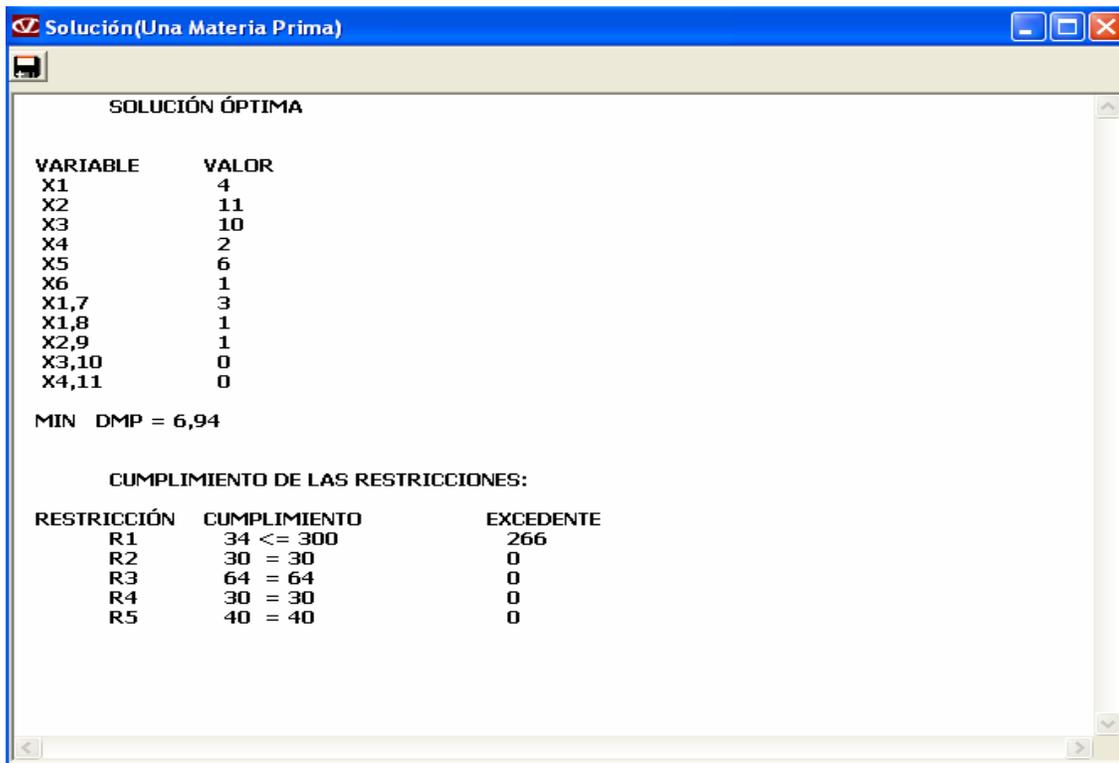


Figura 18. -Caso de uso: Reporte del plan de corte optimizado para una materia prima en existencia-

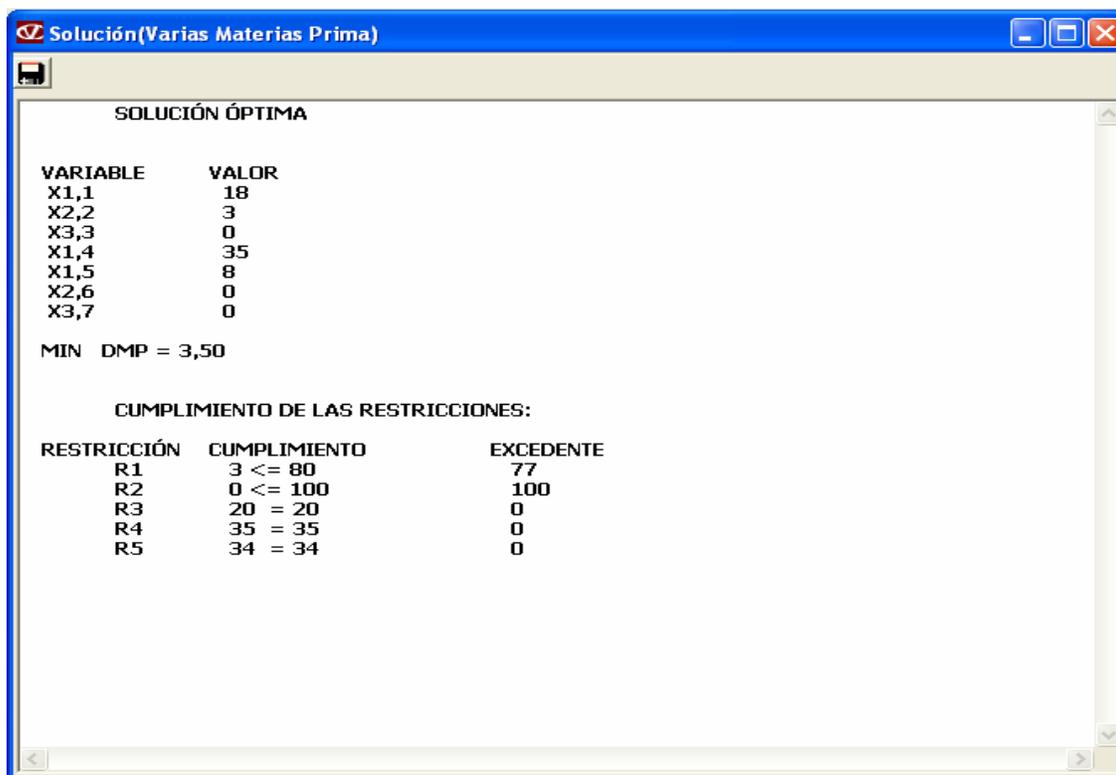


Figura 19. -Caso de uso: Reporte del plan de corte optimizado para varias materias prima en existencia-

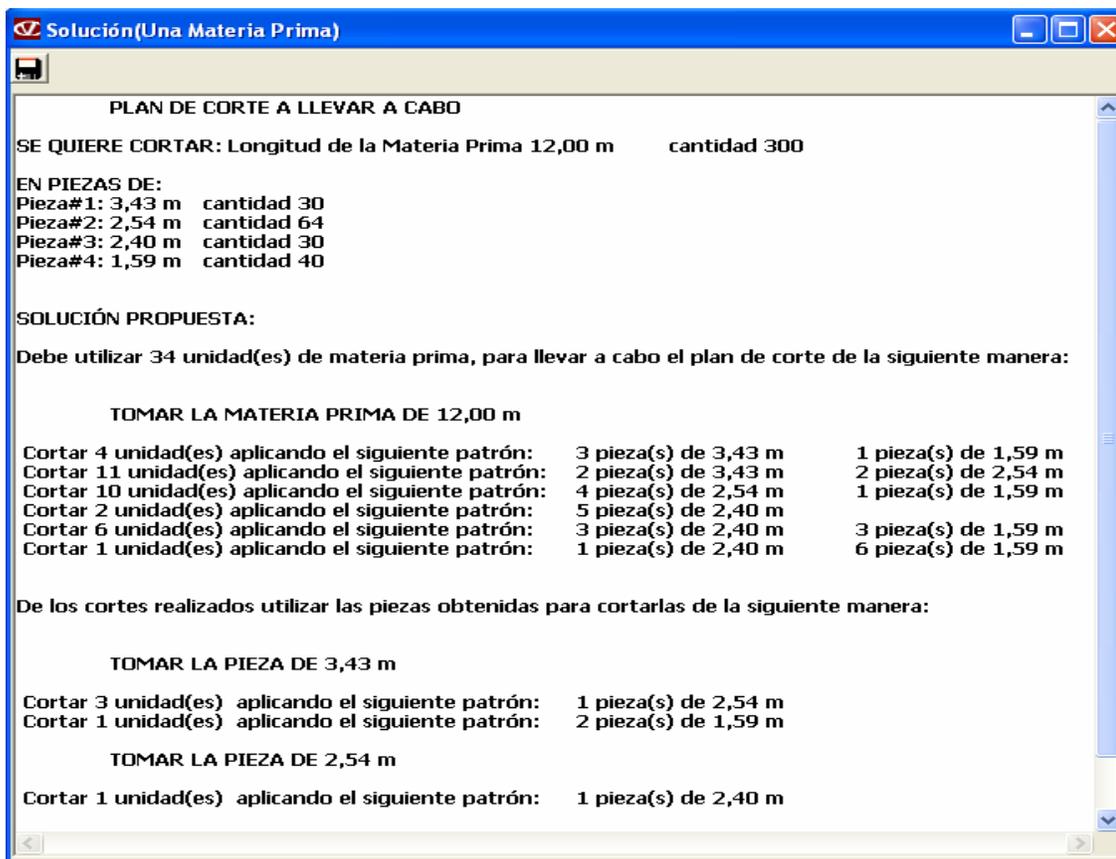


Figura 20. -Caso de uso: Reporte práctico para una materia prima en existencia-

Solución(Varias Materias Prima)

PLAN DE CORTE A LLEVAR A CABO

SE QUIERE CORTAR:
Longitud de la Materia Prima 12,00 m
Longitud de la Materia Prima 10,50 m cantidad 80
Longitud de la Materia Prima 9,00 m cantidad 100

EN PIEZAS DE:
Pieza#1: 3,43 m cantidad 20
Pieza#2: 3,36 m cantidad 35
Pieza#3: 1,70 m cantidad 34

SOLUCIÓN PROPUESTA:

Debe utilizar 18 unidad(es) de materia prima, de longitud 12,00 m para llevar a cabo el plan de corte:
Debe utilizar 3 unidad(es) de materia prima, de longitud 10,50 m para llevar a cabo el plan de corte:

TOMAR LA MATERIA PRIMA DE 12,00 m

Cortar 18 unidad(es) aplicando el siguiente patrón: 3 pieza(s) de 3,43 m 1 pieza(s) de 1,70 m

TOMAR LA MATERIA PRIMA DE 10,50 m

Cortar 3 unidad(es) aplicando el siguiente patrón: 3 pieza(s) de 3,43 m

De los cortes realizados utilizar las piezas obtenidas para cortarlas de la siguiente manera:

TOMAR LA PIEZA DE 3,43 m

Cortar 35 unidad(es) aplicando el siguiente patrón: 1 pieza(s) de 3,36 m
Cortar 8 unidad(es) aplicando el siguiente patrón: 2 pieza(s) de 1,70 m

Figura 21. -Caso de uso: Reporte práctico para varias materias prima en existencia-