

**Universidad de las Ciencias Informáticas**

**Facultad 5 de Entornos Virtuales**



# **ProgProd**

## **Software para la programación de la producción**

**Trabajo de diploma para optar por el título de Ingeniería en Ciencias Informáticas**

**Autor: Miguel Ignacio Bazán Naranjo**

**Tutores:**

**Dra. Rosario Garza Ríos**

**Dra. Ileana Pérez Vergara**

**Ciudad de La Habana, Cuba**

**Julio, 2007**

**Declaración de Autoría**

Declaro que soy el único autor de esta tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del \_\_\_\_\_.

\_\_\_\_\_  
Firma del Autor

Miguel Ignacio Bazán Naranjo

\_\_\_\_\_  
Firma del Tutor

Dra. Rosario Garza Ríos

**Agradecimientos**

*Al Amor y la Amistad que me brindaron mis padres, mi novia, mis familiares y los de mi novia.*

*A mis amistades por darme ánimo para que llegara hasta el final de este trabajo.*

*A mi tutora por haberme hecho de sus complicaciones, un miembro más;  
y dedicarme su tiempo y paciencia en cada momento  
en que desarrollaba este trabajo.*

**Dedicatoria**

*A mis padres Miguel e Idalmis por haberme apoyado en todas las etapas de mi vida, inculcándome valores que un profesional de este país debe tener.*

*A mi novia Ailín por brindarme todo su apoyo y su amor en mis días duros de trabajo y ayudarme a llegar a este momento tan importante de mi vida.*

*A mis familiares por el amor brindado y su preocupación por mí.*

*A los familiares de mi novia los cuales me han apoyado muchísimo para que lograra la culminación de este trabajo.*

*A mis profesores por nutrirme de valores profesionales.*

*A mis amigos por acompañarme en momentos de trabajo y de descanso.*

*A mi patria por haberme permitido llegar a ser quien soñé cuando niño, una semilla más de esta revolución.*

*A mi comandante Fidel Castro por confiar en nosotros.*

**Resumen**

El presente trabajo desarrolla una aplicación informática para la programación de la producción para las empresas cubanas, así como para los centros de enseñanza superior. Existen en la actualidad varios modelos para la planificación y programación de la producción, pero estos no presentan una aplicación informática que pueda ser usada por nuestras empresas cubanas, ya que son destinadas a resolver los problemas existentes en sus propias empresas, es por ello que con este trabajo se quiere desarrollar una aplicación para usos de nuestras empresas y para la enseñanza superior que estudie esta temática. Esta aplicación resolverá problemas para la secuenciación de  $n$  tareas en una máquina, en dos y en tres, y así lograr mediante gráficas, observaciones sobre los tiempos de demora de cada pedido por las máquinas, y demás datos que sean de nuestro interés. Además será capaz de resolver problemas que permitan obtener el balance de líneas de ensamblaje, el cual consiste en asignar las tareas a estaciones de trabajo tal que se optimice un indicador de desempeño determinado, brindando además el tiempo ocioso que existió en la línea y la eficiencia de esta. Todas estas funcionalidades estarán a disposición de la entidad que la desee.

**Palabras claves:** Secuenciación, máquina, tarea, balance, línea, ensamble, estación.

## Índice

<b>INTRODUCCIÓN</b> -----	<b>1</b>
<b>CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA</b> -----	<b>4</b>
1.1 PROGRAMACIÓN DE LA PRODUCCIÓN -----	5
1.1.1 Pronósticos-----	7
1.1.2 Planeación a largo plazo-----	8
1.1.3 Planeación Agregada-----	9
1.1.4 Programa Maestro-----	11
1.1.5 Programación de la Producción -----	13
1.2 MÉTODOS PARA LA PROGRAMACIÓN DE LA PRODUCCIÓN -----	15
1.3 SOFTWARE PARA LA PROGRAMACIÓN DE LA PRODUCCIÓN -----	21
1.3.1 Software de Gestión Industrial (MRP) -----	21
1.3.2 Motor de modelado-----	22
1.3.3 Task Manager -----	22
<b>CAPÍTULO 2 TECNOLOGÍAS ACTUALES A CONSIDERAR</b> -----	<b>24</b>
2.1 FUNDAMENTACIÓN DE LOS LENGUAJES DE PROGRAMACIÓN -----	24
2.1.1 Pascal-----	24
2.1.2 Delphi-----	25
2.1.3 C++ -----	26
2.2 COMPILADOR DE C++-----	27
2.3 FUNDAMENTACIÓN DE LA METODOLOGÍA DE DESARROLLO UTILIZADA-----	28
2.3.1 El Proceso Unificado de Modelado (RUP) -----	28
2.3.2 Lenguaje Unificado de Modelado (UML)-----	31
2.4 TECNOLOGÍAS A EMPLEAR-----	32
<b>CAPÍTULO 3 CARACTERÍSTICAS DEL SISTEMA</b> -----	<b>34</b>
3.1 MODELO DEL DOMINIO-----	34
3.2 DIAGRAMA DE CLASES DEL MODELO DE DOMINIO -----	35
3.3 REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES DEL SISTEMA -----	35
3.3.1 Requerimientos funcionales (RF) -----	35
3.3.2 Requerimientos no Funcionales (RNF).-----	37
3.4 DIAGRAMA DE CASO DE USO DEL SISTEMA-----	38
3.5 DESCRIPCIÓN DE LOS ACTORES Y DIAGRAMA DE LOS CASOS DE USO DEL SISTEMA A AUTOMATIZAR -----	38
3.5.1 Actores del sistema y su justificación-----	38
3.5.2 Descripción de los Casos de Uso del Sistema. -----	39
<b>CAPÍTULO 4 CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA</b> -----	<b>48</b>
4.1 DIAGRAMAS DE CLASES DEL DISEÑO -----	48
4.2 DIAGRAMAS DE INTERACCIÓN -----	53
4.3 DESCRIPCIÓN DE LAS CLASES DEL DISEÑO-----	56
4.4 ESTÁNDARES DE CODIFICACIÓN -----	60
4.5 APARIENCIA DE LA INTERFAZ Y TRATAMIENTO DE ERRORES -----	61
4.6 DIAGRAMA DE DESPLIEGUE -----	64
4.7 DIAGRAMA DE COMPONENTES-----	64

CONCLUSIONES	66
RECOMENDACIONES	67
BIBLIOGRAFÍA	68
REFERENCIAS BIBLIOGRÁFICAS	69
GLOSARIO DE TÉRMINOS	71
ÍNDICE DE FIGURAS	73
ÍNDICE DE TABLAS	74

### Introducción

La emergencia de los nuevos paradigmas tecno-organizativos y la consolidación del proceso de globalización de los mercados, han aumentado el rol que los agentes económicos asignan a las actividades innovativas en la búsqueda de mayor competitividad. Esta revolución tecnológica, centrada en torno a las tecnologías de la información, está modificando la base material de la sociedad a un ritmo acelerado. Las economías de todo el mundo se han hecho interdependientes a escala global, introduciendo una nueva forma de relación entre economía, estado y sociedad.

Se hace necesaria la inserción en un mercado la programación de la producción para un mejor aprovechamiento de los recursos y así mismo una mayor calidad en los productos. Cuando decimos programación de la producción consiste en la fijación de planes y horarios de la producción, de acuerdo a la prioridad de la operación por realizar, determinando así su inicio y fin, para lograr el nivel más eficiente. La función principal de la programación de la producción consiste en lograr un movimiento uniforme y rítmico de los productos a través de las etapas de producción, se inicia con la especificación de lo que debe hacerse, en función de la planeación de la producción; incluye la carga de los productos a los centros de producción y el despacho de instrucciones pertinentes a la operación. También podemos decir que la programación de la producción consiste en asignar las tareas de fabricación a los recursos limitados de la planta a través del tiempo.

En los últimos años se ha estado produciendo un notable incremento de la importancia que tiene el Subsistema de Producción en el desarrollo de la actividad empresarial. Los Sistemas de Gestión de la Producción integran las diferentes funciones de planificación y mando de la producción, a partir de la utilización de técnicas, diagramas, gráficos y software; que facilitan los cálculos y decisiones en torno a la selección de las mejores variantes de producción.

En la actualidad existen diferentes alternativas de Sistemas de Gestión de la Producción (SPCP), acorde a las características propias del proceso productivo (variedad, volumen de producción, complejidad del producto, nivel técnico y tecnológico, etc.), cuyo objetivo es controlar el proceso de producción dentro del sistema empresarial.



A pesar del desarrollo alcanzado por la gestión de la producción en las empresas cubanas, así como los centros de enseñanza superior, estas no cuenta con herramientas computacionales que de forma rápida y flexible permitan obtener la planificación operativa de la producción, entre las que se encuentran obtener la mejor secuenciación de tareas en máquinas y el mejor balance de la líneas de ensamblaje.

Por lo antes expuesto el **problema científico** de este trabajo se basa en como resolver problemas de programación de la producción mediante la secuenciación de tareas en máquinas y balance de líneas de ensamblajes a través de una aplicación informática, teniendo como **objeto de estudio** el proceso de gestión de la producción en empresas cubanas, así como en la enseñanza universitaria en las carreras de ingeniería industrial e ingeniería de procesos agroindustriales para la Tarea Álvaro Reinoso (TAR).

El **campo de acción** estará centrado en la elaboración de una aplicación para la programación de la producción para estas instituciones, por tanto, para resolver este problema se plantea como **hipótesis** la implementación del software ProgProd para la programación de la producción, disponiéndose de una herramienta sencilla y amigable que permitirá a las empresas obtener la programación de los trabajos; para lograr una mejor utilización de los recursos y a los estudiantes les servirá como soporte para resolver estos tipos de problemas en la asignatura de Investigación de Operaciones, lo que redundará en la mejor asimilación de los contenidos garantizando la obtención de mejores resultados docentes.

El **objetivo general** de este trabajo es la implementación de una aplicación informática que permita obtener la secuenciación de tareas en máquinas y el balance de líneas de ensamblaje.

Para dar cumplimiento a estos objetivos se hace necesario cumplir las siguientes **tareas de investigación**:

- Realizar un estudio de los métodos que existen en el mundo para la programación de la producción.
- Hacer un estudio sobre los softwares que existen en la actualidad para la programación de la producción (funcionalidades, posibilidades de adquisición, etc.).
- Seleccionar las herramientas a utilizar para el desarrollo de la aplicación.

La **novedad científica** de este trabajo lo constituye el desarrollo de una aplicación informática que permite realizar la planificación operativa de la producción en los casos en los que el sistema productivo se base, tanto secuenciación de tareas en máquinas, así como en balancear líneas de ensamblaje.

El presente trabajo consta de introducción, 4 capítulos, conclusiones, recomendaciones, bibliografía, referencias bibliográficas, glosario de términos, índice de figuras y tablas:

**Capítulo 1 Fundamentación teórica:** se realiza una reseña desde la planificación de la producción hasta llegar a la última etapa de dicha planificación, la cual es la programación de la producción; donde explica este tema con la claridad que el lector prefiere, además se hace un estudio de los diferentes softwares que existen en la actualidad.

**Capítulo 2 Tendencias y tecnologías actuales:** se hace una descripción de las tendencias y tecnologías actuales posibles de usar en el desarrollo del sistema, y se fundamenta la utilización de las metodologías y tecnologías sobre las que se apoya la propuesta.

**Capítulo 3 Características del sistema:** en este capítulo se presentan los diagramas de clases del dominio, el diagrama de casos de uso del sistema en cuestión; así como los requisitos funcionales y no funcionales que debe cumplir el sistema para su funcionamiento. Esto permite al usuario entender aun más el funcionamiento de la aplicación.

**Capítulo 4 Construcción de la solución propuesta:** en este capítulo se muestran los diagramas de clases del diseño de cada caso de uso y el diagrama general de clases del diseño, los diagramas de interacción de cada caso de uso; además se presenta el diagrama de componentes de dicha aplicación.

## **Capítulo 1 Fundamentación Teórica**

### **Introducción**

En toda empresa industrial la programación de la producción es un problema relevante en el sentido que determina el nivel de cumplimiento que la empresa tendrá con las fechas de entrega de las órdenes a sus clientes. Adicionalmente este es un problema difícil de resolver y que requiere, el uso de herramientas computacionales especializadas. Se han desarrollado innumerables modelos para programar la producción a lo largo de la historia, cada modelo se basa en una serie de hipótesis sobre la tecnología que se utilizan en el proceso, la distribución en planta de las máquinas, los objetivos de la empresa y sus restricciones.

La mayoría de los softwares comerciales utilizados para programar la producción simplifica el sistema real, lo cual sacrifica su aplicabilidad en la empresa; esto redundando además en largos períodos de implantación. Una forma de resolver estos problemas consiste en brindar más flexibilidad al software de forma tal que:

- (i) Capture adecuadamente las características de las plantas de producción.
- (ii) Pueda ser fácilmente configurado de acuerdo a las necesidades de cada empresa.

La programación de la producción señala cuando se necesitan en el trabajo, las máquinas o las instalaciones para elaborar un producto o prestar un servicio, es la etapa de planificación que precede a la producción propiamente dicha. La forma en que se desarrolla la programación es muy distinta en función de cual sea el tipo de sistema productivo por el que se haya decantado la empresa. En industrias de procesamiento, el problema de la asignación de recursos puede resolverse recurriendo a la programación lineal; en la producción masiva, la programación de la producción se ve fuertemente condicionada por la organización de la cadena de montaje. Lo normal es que solamente haya que decidir cual es el ritmo al que circulan los productos por la cadena, y durante cuántas horas diarias se mantiene esta en actividad[1]. En este capítulo se abordará la planificación de la producción, explicando antes la planificación de la misma, ya que este es la que la fundamenta, puesto que es la última etapa de la planificación de la producción.

### 1.1 Programación de la producción

Para el sistema de dirección de las empresas las funciones de planificación, programación y control representan un verdadero contenido científico y práctico; pues son el medio de planteamientos de objetivos y la medida de la eficacia de dicho sistema. La planificación proporciona unas claves de referencia para la toma de decisiones, la cual se concibe en el marco de determinada organización, que permitirá controlar la ejecución de estas. En todo este proceso será preciso elaborar determinadas previsiones económicas a corto, medio y largo plazo para apoyar adecuadamente las decisiones empresariales.

Existen varios y similares enfoques con respecto al proceso de planificación, programación y control de la producción que han sido tratados por diversos autores entre los que están: Vollmann et al [1997] y Domínguez Machuca et al [1995], que dicen que el proceso de planificación y control de la producción debe seguir un enfoque jerárquico, en el que se logre una integración vertical entre los objetivos estratégicos, tácticos y operativos y además se establezca su relación horizontal con las otras áreas funcionales de la compañía.

**Básicamente las cinco fases que componen el proceso de planificación y control de la producción son: [Domínguez Machuca 1995]:**

1. Planificación estratégica o a largo plazo.
2. Planificación agregada o a medio plazo.
3. Programación maestra.
4. Programación de componentes.
5. Ejecución y control.

Es importante anotar, que de acuerdo con Domínguez Machuca [1995], estas fases se deberán llevar a cabo en cualquier empresa manufacturera, independientemente de su tamaño y actividad, aunque la forma en que estas se desarrollen dependerá de las características propias de cada sistema productivo.

La figura 1, resume las principales fases mencionadas junto con los planes que de ellos se derivan, relacionando por un lado, los niveles de planificación empresarial y por otro la planificación y gestión de la capacidad.

Teniendo en cuenta los aspectos que se deben considerar en el proceso de planificación, programación y control de la producción y en aras de su importancia en las acciones de mejoramiento de la capacidad competitiva de una organización, a continuación se procederá a analizar de manera detallada los aportes de distintos autores en cuanto a conceptos, métodos y técnicas más empleados en cada una de sus fases.

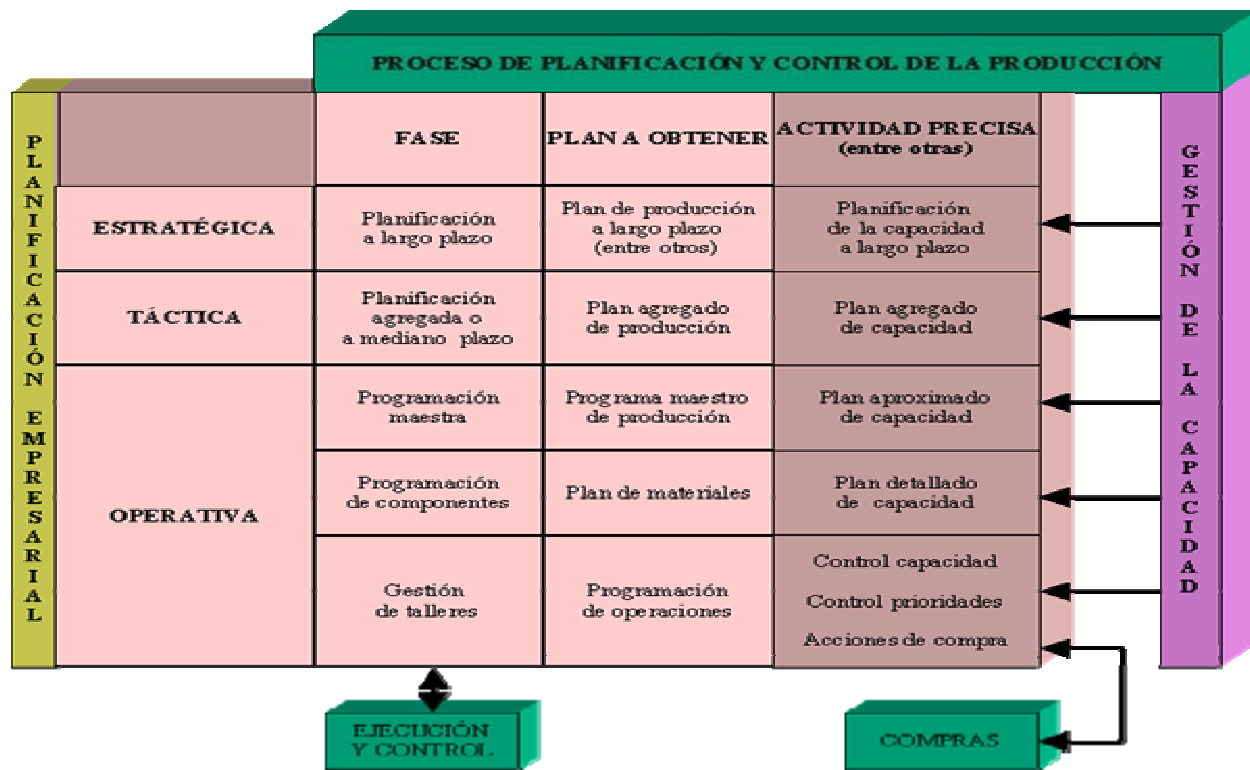


Figura 1. Proceso de Planificación, programación y control de la producción

### 1.1.1 Pronósticos

En aproximación a lo expresado por Riggs [1998], Domínguez Machuca et al [1995], Buffa & Sarin [1992], Adam & Ebert [1991]; Hanke & Deitsch [1996] y Voris [1977], se puede afirmar que los pronósticos son el primer paso dentro del proceso de planificación de la producción y sirven como punto de partida, no solo para la elaboración de los planes estratégicos, sino además, para el diseño de los planes a mediano y corto plazo, lo cual permite a las organizaciones visualizar de manera aproximada los acontecimientos futuros, eliminar en gran parte la incertidumbre y reaccionar con rapidez a las condiciones cambiantes con algún grado de precisión.

Desde el punto de vista conceptual, algunos autores [Tawfik & Chauvel, 1992; Adam & Ebert, 1991; Kalenatic & Blanco, 1993] expresan la importancia de diferenciar entre los términos predicción y pronóstico, ya que de acuerdo a su criterio, las predicciones se basan en la consideración de aspectos subjetivos dentro del proceso de estimación de eventos futuros; mientras que los pronósticos, se desarrollan a través de procedimientos científicos, basados en datos históricos que son procesados mediante métodos cuantitativos.

En lo referente a los tipos de pronósticos, estos pueden ser clasificados de acuerdo a tres criterios: según el horizonte de tiempo, según el entorno económico abarcado y según el procedimiento empleado [Hanke & Deitsch, 1996]. Los pronósticos según el horizonte de tiempo pueden ser de largo plazo, mediano plazo o corto plazo [Domínguez Machuca et al, 1995; Lockyer, 1995; Hanke & Deitsch, 1996] y su empleo va desde la elaboración de los planes a nivel estratégico hasta los de nivel operativo. Los pronósticos según el entorno económico pueden ser de tipo micro o de tipo macro y se definen de acuerdo al grado en que intervienen pequeños detalles vs. grandes valores resumidos [Hanke & Deitsch, 1996].

Los pronósticos según el procedimiento empleado pueden ser de tipo puramente cualitativo, en aquellos casos en que no se requiere de una abierta manipulación de datos y solo se utiliza el juicio o la intuición de quien pronostica o puramente cuantitativos, cuando se utilizan procedimientos matemáticos y estadísticos que no requieren los elementos del juicio.

### 1.1.2 Planeación a largo plazo

La estrategia de operaciones debe dar como resultado un patrón consistente de toma de decisiones en las operaciones y una ventaja competitiva para la compañía. Así mismo, Chase & Aquilano [1995], expresan como aspecto importante a considerar, que dicha estrategia debe especificar la manera en que la empresa empleará sus capacidades productivas para apoyar la estrategia corporativa. Todo esto significa, que la estrategia de operaciones debe surgir de una estrategia empresarial a largo plazo y a su vez, debe integrarse de manera horizontal con las estrategias de los demás subsistemas de la compañía.

De acuerdo con esta afirmación y en concordancia con Domínguez Machuca et al [1995], la estrategia de operaciones se constituye como un plan a largo plazo para el subsistema de operaciones, en el que se recogen los objetivos a lograr y los cursos de acción, así como la asignación de recursos a los diferentes productos y funciones. Todo ello debe perseguir el logro de los objetivos globales de la empresa en el marco de su estrategia corporativa, constituyendo además un patrón consistente para el desarrollo de las decisiones tácticas y operativas del subsistema. Lo anterior, no difiere del concepto de Schroeder [1992], quien agrega además que la estrategia de operaciones debe ser una estrategia funcional que debe guiarse por la estrategia empresarial y cuyo corazón debe estar constituido por la misión, la competencia distintiva, los objetivos y las políticas.

**En consonancia con lo anterior, Domínguez Machuca et al [1995] plantea, que las dos funciones básicas que ha de cumplir la estrategia de operaciones son:**

1. Servir como marco de referencia para la planificación y control de la producción, de la cual es su punto de partida.
2. Marcar las pautas que permitan apreciar en qué medida el subsistema de operaciones está colaborando el logro de la estrategia corporativa.

**Dentro de este propósito, las decisiones básicas que deben ser contempladas en la estrategia de operaciones son:**

1. Decisiones de posicionamiento que afectan la dirección futura de la compañía y dentro de la cual se incluyen los objetivos a largo plazo, el establecimiento de las prioridades competitivas, la fijación del modelo de gestión de la calidad, la selección de productos y la selección de procesos.
2. Decisiones de diseño, concernientes al subsistema de operaciones que implican compromiso a largo plazo y entre las cuales se encuentran el diseño del productos y procesos, la mano de obra, la apropiación de nuevas tecnologías, decisiones de capacidad, localización y distribución de instalaciones y sistemas de aprovisionamiento.

### 1.1.3 Planeación Agregada

La planeación agregada denominada también planeación combinada [Meredith & Gibbs, 1986], se encuentra ubicada en el nivel táctico del proceso jerárquico de planeación y tiene como misión fundamental, en aproximación al planteamiento de varios autores [Schroeder,1992; Chase & Aquilano,1995; Nahmias,1997; Heizer & Render,1997; Rusell & Taylor,1998; Domínguez Machuca et al, 1995], la de establecer los niveles de producción en unidades agregadas a lo largo de un horizonte de tiempo que, generalmente, fluctúa entre 3 y 18 meses, de tal forma que se logre cumplir con las necesidades establecidas en el plan a largo plazo, manteniendo a la vez niveles mínimos de costos y un buen nivel de servicio al cliente.

El término agregado, en este nivel de planeación, implica que las cantidades a producir se deben establecer de manera global o como lo expresa Schroeder [1992] para una medida general de producción o cuando mucho para algunas pocas categorías de productos acumulados. De acuerdo con Nahmias [1997], puede ser aconsejable utilizar unidades agregadas tales como familias de productos, unidad de peso, unidad de volumen, tiempo de uso de la fuerza de trabajo o valor en dinero. De todas maneras, cualquier unidad agregada que se escoja debe ser significativa, fácilmente manejable y comprensible dentro del plan.

De otra parte, dentro del proceso de elaboración del plan agregado y en aras del cumplimiento de su objetivo fundamental, es importante el manejo de las variables que pueden influir en este, las cuales pueden ser clasificadas en dos grandes grupos [Schroeder, 1992]: En primer lugar, están las variables de



oferta, las cuales permiten modificar la capacidad de producción a través de la programación de horas extras, contratación de trabajadores eventuales, subcontratación de unidades y acuerdos de cooperación; en segundo lugar, están las variables de demanda, las cuales pueden influir en el comportamiento del mercado mediante la publicidad, el manejo de precios, promociones, etc. Así mismo, existen varias estrategias para la elaboración del plan agregado, las cuales han sido clasificadas por la mayoría de los autores en dos grupos, subdivididos así: [Schroeder, 1992; Chase & Aquilano, 1995; Nahmias, 1997; Heizer & Render, 1997; Rusell & Taylor, 1998; Vollmann et al, 1997; Domínguez Machuca et al, 1995]:

**Estrategias puras:** Mano de obra nivelada (con empleo de horas extras o trabajadores eventuales)

Estrategia de persecución, adaptación a la demanda o de caza: (con o sin empleo de la subcontratación).

**Estrategias mixtas:** Se realizan mezclando varias estrategias puras.

Debido a las diferentes estrategias que se pueden adoptar, se debe obtener un plan que satisfaga las restricciones internas de la organización y a la vez mantenga el costo de utilización de los recursos lo más bajo posible.

**En cuanto a las técnicas existentes en la elaboración de planes agregados, de acuerdo con los autores consultados (Ibídem), las más renombradas son las siguientes:**

1. Métodos manuales de gráficos y tablas
2. Métodos matemáticos y de simulación: programación lineal (método simplex y método del transporte), programación cuadrática, simulación con reglas de búsqueda (Search Decision Rules) y programación con simulación.
3. Métodos heurísticos: método de los coeficientes de gestión, método PSH (Production Switching Heuristic), reglas lineales de decisión (LDR) y búsqueda de reglas de decisión (SDR).

### 1.1.4 Programa Maestro

Una vez concluido el plan agregado, el siguiente paso consiste en traducirlo a unidades o ítems finales específicos. Este proceso es lo que se conoce como desagregación [Dominguez Machuca, 1995], subdivisión [Adam & Ebert, 1991] o descomposición [Narasimhan et al, 1996] del plan agregado y su resultado final se denomina programa maestro de producción (Master Production Schedule, MPS). Básicamente, se puede afirmar que un programa maestro de producción, es un plan detallado que establece la cantidad específica y las fechas exactas de fabricación de los productos finales [Heizer & Render, 1997; Russell & Taylor, 1998]. Al respecto, Vollmann et al [1997] agrega que un efectivo MPS debe proporcionar las bases para establecer los compromisos de envío al cliente, utilizar eficazmente la capacidad de la planta, lograr los objetivos estratégicos de la empresa y resolver las negociaciones entre fabricación y marketing.

**Las unidades en que puede ser expresado un MPS son: [Heizer & Render, 1997]**

1. Artículos acabados en un entorno continuo. (Make to stock).
2. Módulos en un entorno repetitivo (Assemble to stock).
3. Pedido de un cliente en un entorno de taller (Make to order).

En cuanto al horizonte de tiempo de un MPS, la mayoría de los autores coinciden en que este puede ser variable y que dependiendo del tipo de producto, del volumen de producción y de los componentes de tiempo de entrega, este puede ir desde unas horas hasta varias semanas y meses, con revisiones, generalmente, semanales.

**Así mismo, Chase & Aquilano [1995], agregan que en aras de mantener el control y evitar el caos en el desarrollo del MPS, subdivide el horizonte de tiempo en tres marcos:**

1. **Fijo:** Periodo durante el cual no es posible hacer modificaciones al PMP.
2. **Medio fijo:** Aquel en el que se pueden hacer cambios a ciertos productos.
3. **Flexible:** Lapso de tiempo más alejado, en el cual es posible hacer cualquier modificación al MPS.

En lo referente a los insumos para la obtención del MPS es importante la consideración de los siguientes elementos [Domínguez Machuca et al, 1995]: el plan agregado en unidades de producto, las previsiones de ventas a corto plazo en unidades de producto, los pedidos en firme comprometidos con los clientes, la capacidad disponible de la instalación o el centro de trabajo y por último, otras fuentes de demanda.

**Dentro del proceso de formalización del MPS existen algunas de las funciones claves que este debe cumplir son [Monks, 1991]:**

1. Traducir los planes agregados en artículos finales específicos.
2. Evaluar alternativas de programación.
3. Generar requerimientos de materiales.
4. Generar requerimientos de capacidad y maximizar su utilización.
5. Facilitar el procesamiento de la información.
6. Mantener las prioridades válidas.

Con respecto a las técnicas existentes para desagregar el plan agregado y traducirlo a un MPS, se han desarrollado algunos modelos analíticos [Monks, 1991; Domínguez Machuca, 1995; Schroeder, 1992; Narasimhan et al, 1996] y de simulación los cuales, a juicio de los autores citados, adolecen de los mismos problemas de la planificación agregada, siendo los de mayor uso por parte de los empresarios, los métodos de prueba y error.

**No obstante, Narasimhan et al [1996], plantea la existencia de otros métodos para la desagregación, a saber:**

- **Método de corte y ajuste:** Pone a prueba diversas distribuciones de la capacidad para los productos en un grupo hasta que se determine una combinación satisfactoria.
- **Métodos de programación matemática:** Modelos de optimización que permiten la minimización de los costos.

- **Métodos heurísticos:** Al igual que en la planeación agregada, permiten llegar a soluciones satisfactorias aunque no óptimas.

**Por último y de acuerdo con Vollmann [1997], es importante anotar que un buen MPS debe tomar en cuenta las limitaciones de capacidad y mantenerse factible desde este punto de vista, lo cual puede lograrse aplicando las siguientes técnicas:**

- Planificación de capacidad usando factores agregados (CPOF, Capacity Planning Using Overall Factors).
- Listas de capacidad (Capacity Bills).
- Perfiles de recursos (Resource profiles). [2]

### 1.1.5 Programación de la Producción

La programación de la producción consiste en asignar las tareas de fabricación a los recursos limitados de la planta a través del tiempo. El programa de producción trata, por tanto, de ordenar en el tiempo la fabricación de los distintos lotes establecidos en el Plan Maestro y asignar los recursos que se utilizan para este cometido. El resultado de la programación es un listado de las tareas a realizar en cada máquina (órdenes de trabajo). Cada una con su tiempo planificado de comienzo y de finalización. También podemos decir que la meta en programar la producción es el integrar todas las tareas de la producción de modo que ninguna esperar innecesaria ocurra. [3]

Este último paso dentro del proceso jerárquico de planificación y control, permitirá saber a cada trabajador o a cada responsable de un centro de trabajo lo que debe hacer para cumplir el plan de materiales y con el plan agregado y los planes estratégicos de la empresa. [Domínguez Machuca et al, 1995]. Estas actividades, se enmarcan dentro de la fase de ejecución y control, que en el caso de las empresas fabriles se denomina gestión de talleres. Un taller de trabajo, de acuerdo con Chase & Aquilano [1995], se define como una organización funcional cuyos departamentos o centros de trabajo se organizan alrededor de ciertos tipos de equipos u operaciones; en ellos, los productos fluyen por los departamentos en lotes que corresponden a los pedidos de los clientes.

Es importante dentro de esta fase de gestión, tomar en consideración el tipo de configuración productiva que tiene el taller, pues dependiendo de esta, así mismo será la técnica o procedimiento a emplear en su programación y control.

**El cumplimiento de estas actividades debe responder a las siguientes preguntas del programador [Schroeder, 1992]:**

1. ¿Qué capacidad se necesita en el centro de trabajo?
2. ¿Qué fecha de entrega se debe prometer en cada pedido?
3. ¿En qué momento comenzar cada pedido?
4. ¿Cómo asegurar que los pedidos terminen a tiempo?

Las pregunta 1 puede ser resuelta a través de los análisis de carga; las preguntas 2 y 3 se resuelven con la aplicación de las técnicas de Secuenciación y la programación detallada y la pregunta 4 con el análisis de fluidez y el control insumo producto. [2]

**La programación de la producción tiene como objetivos lograr:**

1. Cumplir con las fechas de entrega a los clientes.
2. Minimizar los retrasos en el trabajo.
3. Minimizar el tiempo de respuesta.
4. Minimizar la duración del ciclo de fabricación.
5. Maximizar la utilización de los recursos (máquinas o mano de obra).
6. Minimizar los tiempos de espera en el proceso.
7. Minimizar el inventario de productos en curso.

**La programación de la producción es afectada por:**

- **Materiales:** Para cumplir con las fechas comprometidas para su entrega.
- **Capacidad del personal:** Para mantener bajos costos al utilizarlo eficazmente, en ocasiones afecta la fecha de entrega.
- **Capacidad de producción de la maquinaria:** Para tener una utilización adecuada de ellas, deben observarse las condiciones ambientales, especificaciones, calidad y cantidad de los materiales, la

experiencia y capacidad de las operaciones en aquellas.

- **Sistemas de producción:** Realizar un estudio y seleccionar el más adecuado, acorde con las necesidades de la empresa.

**La función de la programación de la producción tiene como finalidad lo siguiente:**

- Prever las pérdidas de tiempo o las sobrecargas entre los centros de producción.
- Mantener ocupada la mano de obra disponible.
- Cumplir con los plazos de entrega establecidos.[4]

### 1.2 Métodos para la programación de la producción

**Pueden ser de dos tipos según [Mayer, 1977; Domínguez Machuca et al, 1995; Adam & Ebert, 1991; Chase & Aquilano, 1995; Nahmias, 1997; Tawfik & Chauvel, 1992]:**

1. **Talleres de configuración continua o en serie:** Aquellos en donde las máquinas y centros de trabajo se organizan de acuerdo a la secuencia de fabricación (líneas de ensamblaje), con procesos estables y especializados en uno o pocos productos y en grandes lotes. En ellos, las actividades de programación están encaminadas principalmente, a ajustar la tasa de producción periódicamente.

Este problema de diseño para encontrar formas para igualar los tiempos de trabajo en todas las estaciones se denomina problema de balanceo de línea. En el cual deben existir ciertas condiciones para que la producción en línea sea práctica:

- **Cantidad:** El volumen o cantidad de producción debe ser suficiente para cubrir el costo de la preparación de la línea. Esto depende del ritmo de producción y de la duración que tendrá la tarea.
- **Equilibrio:** Los tiempos necesarios para cada operación en línea deben ser aproximadamente iguales.

- **Continuidad:** Deben tomarse precauciones para asegurar un aprovisionamiento continuo del material, piezas, subensambles, etc., y la prevención de fallas de equipo.

### Los casos típicos de balanceo de línea de producción son:

- Conocidos los tiempos de las operaciones, determinar el número de operarios necesarios para cada operación.
- Conocido el tiempo de ciclo, minimizar el número de estaciones de trabajo.
- Conocido el número de estaciones de trabajo, asignar elementos de trabajo a la misma.

2. **Talleres de configuración por lotes:** En los que la distribución de máquinas y centros de trabajo, se organizan por funciones o departamentos con la suficiente flexibilidad para procesar diversidad de productos. Estos pueden ser de dos tipos[Bera,1996]:

- **Configurados en Flow Shop:** Donde los distintos productos siguen una misma secuencia de fabricación.
- **Configurados en Job Shop:** Aquellos donde los productos siguen secuencias de fabricaciones distintas.

Así mismo, en la práctica, muchos talleres debido a las necesidades de fabricación y exigencias competitivas del mercado actual, han adoptado configuraciones híbridas, de las cuales, la más generalizada es la configuración celular o células de manufactura. Estas constituyen un sistema de fabricación diseñado para procesar familias de piezas, con una distribución física tal, que permite simplificar los procedimientos de planificación y control. [Vollmann, 1997]. En términos generales y en el caso más complejo, las actividades que se presentan en la programación y control de operaciones son [Domínguez Machuca et al, 1995; Schroeder, 1992; Chase & Aquilano, 1995]: Asignación de cargas, Secuenciación de pedidos y programación detallada. A estas, Adam & Eber t [1991], agregan otras dos:

- **Fluidez:** Permite verificar que los tiempos planeados se cumplan, de tal forma que, si existen desviaciones en la producción real, se puedan tomar medidas correctivas a tiempo. [Adam & Ebert, 1991].
- **Control de insumo / Producto:** Controlan los niveles de utilización de la capacidad de cada centro de trabajo, mediante los informes de entrada/salida [Ibídem].

**Asignación de carga:** En aproximación a los conceptos de Heizer & Render [1997], Adam & Ebert [1991], Lockyer [1995], Schroeder [1992] y Domínguez Machuca et al [1995], esta se define como la asignación de tareas a cada centro de trabajo o de proceso, que permite controlar la capacidad y la asignación de actividades específicas en cada centro de trabajo. En general las técnicas más empleadas en la asignación de carga son: Gráficos Gantt, perfiles de carga o diagramas de carga, métodos optimizadores (algoritmo de Kuhn o método Húngaro) y soluciones heurísticas (método de los índices).

**Secuenciación de pedidos:** Esta actividad consiste, en la determinación del orden en que serán procesados los pedidos en cada centro de trabajo, una vez establecida la existencia de capacidad. [Ibídem]. El problema de la secuenciación se hace más complejo en la medida que aumenta el número de centros de trabajo, sin importar la cantidad de pedidos; así mismo, es importante tomar en cuenta el tipo de configuración del taller, pues de esto depende la aplicabilidad de las diferentes técnicas.

En lo referente a talleres configurados en Flow Shop, las técnicas más conocidas son:

- **Técnicas de Secuenciación en una máquina:** algoritmo húngaro, algoritmo de Kauffman, regla SPT y el método de persecución de objetivos utilizado en los sistemas Kanban.
- **Técnicas de Secuenciación en varias máquinas:** regla de Johnson para N pedidos y dos máquinas, regla de Johnson para N pedidos y tres máquinas y reglas para N pedidos y M máquinas (algoritmo de Campbell-Dudek-Schmith, algoritmo de Bera, técnicas de simulación, sistemas expertos y más recientemente los Sistemas Cooperativos Asistidos).



Para los talleres configurados en Job Shop, debido a la diversidad en la secuencia de operaciones, no es posible emplear alguna técnica de optimización, por lo cual la secuencia de operaciones se establece en función de los objetivos específicos de cada programador, a través del uso de reglas de prioridad.[Adam & Ebert, 1991].

Una recopilación realizada en las obras de varios autores, permite determinar que las reglas de prioridad más empleadas son [Buffa & Sarin, 1995; Tawfik & Chauvel, 1992; Monks, 1991; Russell & Taylor, 1998; Mayer, 1977; Domínguez Machuca et al, 1995; Adam & Ebert, 1991; Chase & Aquilano, 1995; Nahmias, 1997; Schroeder, 1992]:

- **FCFS:** First come/ First serve (primero en llegar, primero en ser atendido).
- **SPT:** Shortes Processing Time (menor tiempo de procesamiento).
- **EDD:** Earliest Due date (fecha de entrega más próxima).
- **CR:** Critical Ratio (razón critica o ratio crítico). [2]

### Existen ocho tipologías de sistemas o configuraciones productivas bien definidas:

1. **Configuración por Proyecto:** Producción generalmente de productos únicos de cierta complejidad que requieren gran cantidad de inputs. Estos deben fabricarse en un lugar definido debido a que es difícil o casi imposible transportarlos una vez terminados; como resultado, y a diferencia de cualquier otro proceso productivo, los recursos que comprende deben trasladarse al lugar de operación, ya que aquí no existe flujo del objeto de trabajo, sino que son los recursos técnicos y humanos quienes acuden al lugar de trabajo. Las actividades y recursos se gestionan como un todo, su coordinación adquiere carácter crítico. Existe un connotado interés por el control de los costos y las fechas de terminación.
2. **Configuración de Taller (Job-shop):** El sistema de producción Job-Shop fabrica muchos productos diferentes en volúmenes que varían entre la unidad y pocas unidades de cada producto. Consiste en una fabricación no en serie, de lotes pequeños, para pedidos únicos o de pequeñas cantidades. Por

lo regular implica productos adaptados, diseñados a la medida del cliente y de naturaleza muy poco repetitiva. Se requieren operaciones poco especializadas, las cuales son realizadas por un mismo obrero o por un grupo pequeño de ellos, los cuales tienen la responsabilidad de terminar todo o casi todo el producto. Como se fabrican productos muy diferentes, los recursos son flexibles y versátiles. El flujo material es irregular, aleatorio y varía considerablemente de un pedido al siguiente. Se requiere que el fabricante interprete el diseño y las especificaciones del trabajo, así como que aplique capacidades del alto nivel en el proceso de conversión. En la producción Job-Shop lo que se trata es de obtener un "producto a medida" del cliente.

- 3. Configuración por Lotes:** El sistema de flujo en lotes produce menos variedad de producto en volúmenes más elevados que el caso anterior. El mayor volumen se debe a un aumento de la repetitividad en ciertos artículos que se hacen dominantes. Estos productos se fabrican en lotes, que representan unos pocos meses de requerimientos de clientes. En este caso se requieren más operaciones, y éstas son más especializadas, por lo que difícilmente un mismo operario pueda dominarlas todas con una eficiencia aceptable. En tal sentido, el trabajo se divide en diferentes etapas tecnológicas, en las cuales los lotes sufren distintas operaciones. Así la instalación se suele dividir en secciones o talleres, en los cuales se agrupan los equipos con funciones similares. Se suele emplear una combinación de layouts celulares y funcionales. Los layouts celulares se utilizan cuando es efectivo en cuanto a costos disponer el equipo en células, para producir familias de productos. Como hay muchos productos, el equipo y utillaje son mayormente flexibles, de propósito general. El flujo material es desconectado aunque regular, variable de un pedido a otro, aunque existen pautas de flujo para familias de productos y para grandes lotes, es el sistema más utilizado.
- 4. Configuración en Línea Acompasada por el Equipo (LAE):** El equipo y los procesos están organizados en una línea o líneas especializadas para producir un pequeño número de productos diferentes o familias de productos. Estos sistemas se usan sólo cuando el diseño del producto es estable y el volumen es lo suficientemente elevado para hacer un uso eficiente de una línea especializada con capacidades dedicadas. Se fabrica a una tasa constante, con un flujo automatizado e intensivo en capital. Los operarios realizan tareas relativamente simples a un ritmo determinado por la velocidad de la línea. El control del ciclo productivo está automatizado, existe alta estandarización y una elevada eficiencia en todo el proceso.

5. **Configuración en Línea Acompasada por Operarios (LAO):** Se utiliza cuando el número de productos diferentes es demasiado elevado y los volúmenes de producción demasiado variables para el sistema en línea con flujo acompasado por el equipo. En este sistema, la línea es más flexible que en el caso anterior, y puede funcionar con una variedad de velocidades. La tasa de producción depende del producto particular que se fabrique, del número de operarios asignados a la línea y de la eficacia del trabajo en equipo de los operarios. Aunque los productos sean algo diferentes, son técnicamente homogéneos, usando la misma instalación, personal y la misma secuencia de estaciones de trabajo, aunque alguno de ellos pueda no pasar por alguna que no le es necesaria. El ciclo de productivo está controlado por los operarios a diferencia de la LAE donde dicho control está automatizado, esto hace que sea más flexible y versátil que el anterior.
6. **Configuración de Flujo Continuo:** Este sistema es similar al de línea en flujo acompasado por el equipo. Sin embargo, es más automatizado, más intensivo en capital y menos flexible. Cada máquina y equipo están diseñados para realizar siempre la misma operación y preparados para aceptar de forma automática el trabajo suministrado por la máquina precedente. Está diseñado para fabricar un producto o una familia limitada de productos en volúmenes muy elevados. El diseño del producto es muy estable, a menudo es un producto genérico o «commodity». El flujo material es continuo sincronizado, integrado a través de toda la instalación como si fuera un gran proceso tecnológico. Este rígido sistema, se basa en un proceso muy automatizado, costoso y especializado en la obtención de un producto estándar, donde la homogeneidad es total y absoluta, funcionando continuamente con mínima intervención del personal de línea. Generalmente precisa laborar las 24 horas para procurar ser un sistema costeable y eficiente.
7. **Sistema de Producción JIT:** Es importante distinguir entre el sistema de producción JIT y las técnicas JIT. Las técnicas denominadas JIT incluyen el control estadístico de la calidad, reducción de los tiempos de cambio de útiles (SMED), polivalencia de los trabajadores, versatilidad de los equipos, estandarización de operaciones, el enfoque de la producción mediante «arrastre» (Kanban), layout celular, mantenimiento autónomo, implicación de todo el personal en las decisiones gerenciales, resolución continua de problemas control automático de defectos, etc. Estas técnicas se usan en el sistema de producción JIT, pero también se usan en otros sistemas. El sistema de producción JIT es mucho más que un agregado de técnicas JIT. Surgido en Toyota Motor Co., es un sistema de flujo lineal (virtual o físico) que fabrica muchos productos en volúmenes bajos a medios. Por su diseño, el

sistema JIT fuerza la eliminación de todos los innecesarios ("desperdicios"), y a partir de aquí, impone la mejora continua. Esto conduce naturalmente a costos inferiores, mejoras en la calidad y entregas más rápidas. El sistema JIT es el más difícil de diseñar, implantar y gestionar de todos, y pueden existir diferentes niveles de implantación del mismo.

8. **Sistema Flexible de Fabricación (FMS):** El sistema FMS consiste en un grupo de máquinas controladas por computadoras y sistemas automáticos de manejo, carga y descarga de material, todo ello controlado por un computador supervisor. Un FMS puede funcionar sin atención de personal durante largos periodos. Las máquinas, el sistema de manipulación de materiales y las computadoras son muy flexibles, versátiles, lo que permite a un sistema FMS fabricar muchos productos diferentes en bajos volúmenes. Por ser sumamente costoso, se emplea comúnmente en situaciones en las que no pueden utilizarse sistemas de producción en línea de flujo más simples y baratos. Por lo general, se desarrolla en un entorno CIM (manufactura integrada por computador). [5]

### 1.3 Software para la programación de la producción

A continuación se presentan tres softwares que realizan la programación de la producción.

#### 1.3.1 Software de Gestión Industrial (MRP)

Este software es diseñado para satisfacer las necesidades de las Pequeñas y Medianas Empresas. El mismo se presenta en sus dos modalidades adaptables a sus requerimientos. Además tiene como características, controlar el Stock (Por Partida, Lote y Número de Serie), posee una fórmula de Producción, un costos Standard por Producto, un Ingreso de Producción con deducción automática de Materiales a través de la Fórmula del Producto, calcula los materiales a comprar, emite de Ordenes de Producción (O.P.), emite de O.P. por Pedidos de Clientes, asigna y reserva los materiales por O.P. [6]

### 1.3.2 Motor de modelado

Este motor de modelado permitirá a las pequeñas y medianas empresas programar detalladamente su producción a bajo costo y con alta efectividad. Dicho software permite programar cada una de las actividades dentro de la empresa, así como calcular los tiempos invertidos por cada operario, la intensidad de uso que debe dársele a cada máquina y la cantidad de producto que la fábrica puede entregar a diario, en la semana o en el mes. El motor de modelado de la producción cuenta con dos niveles, uno que permite ingresar en una base de datos: información relacionada con los equipos, operarios, productos, procesos de producción y operaciones de manufactura, y otro denominado capa de aplicación, en el que se elaboran consultas para la toma de decisiones.

La flexibilidad de esta innovación para las empresas está dada por la posibilidad de personalizar el modelo de producción de cada planta, sin importar si pertenece a sectores tan distintos como el de industrias gráficas, alimentos, textiles o metalmecánica. Dicha operación de modelado se lleva a cabo a través de un lenguaje estándar para referirse a elementos como máquinas, materias primas, tiempos, funciones y cargos.

Tanto los académicos de la Universidad de Los Andes, como los ingenieros de Sistemas Corporativos S.A. esperan que con esta innovación las pequeñas y medianas empresas del país logren programar de forma más organizada y sencilla sus pedidos y órdenes de producción para cumplir con los objetivos de entrega. [7]

### 1.3.3 Task Manager

Es un Sistema de gestión de tareas y actividades mediante la captura de datos en planta, basado en terminales de captura de datos programables que se adaptan a cada secuencia de producción; admite el registro de empleados, órdenes de fabricación, tareas y tiempos mediante tarjetas de banda magnética, códigos de barras, tarjetas de proximidad, etc.

**El software de gestión Task Manager como solución para el control de tareas le permite:**

- Gestionar Órdenes de Trabajo y las Tareas asociadas

- Integrarlas con el software Control Horario
- Obtener el cálculo de tiempos muertos
- Conseguir información inmediata de su planta de producción
- Comparar la información de Control Horario y Control de tareas conjuntamente
- Enlazar con su ERP [8]

Después de haber visto las funcionalidades que nos brindan estos softwares podemos decir que tienen limitaciones para las empresas cubanas, ya que muchas de ellas no tienen la tecnología para acceder a estos mediante Internet; además por el costo para adquirirlo se le imposibilita a estas entidades tener estas herramientas. Por lo cual este trabajo se propone desarrollar una aplicación informática que sirva de beneficio a nuestro país.

### Conclusiones

Del estudio bibliográfico realizado se obtienen las siguientes conclusiones:

- La programación de la producción es la etapa dentro de la planificación, que prevé las pérdidas de tiempo o las sobrecargas entre los centros de producción, manteniendo ocupada la mano de obra disponible y cumpliendo con los plazos de entrega establecidos.
- Los softwares que existen en el mundo que están relacionados con este tema no pueden ser adquiridos debido a las limitaciones económicas y tecnológicas que presenta nuestra patria socialista.
- Queda evidenciada la necesidad de la implementación de una aplicación informática que permita resolver estos problemas para que sea utilizada en las empresas cubanas.

# Capítulo 2 Tecnologías Actuales a Considerar

## Introducción

En el presente capítulo se analizan las tecnologías actuales a considerar para desarrollar la aplicación propuesta, en cuanto a sus características, las ventajas y desventajas que presentan las mismas, ya que su utilización permitirá el éxito de la aplicación; además se justifica el lenguaje de programación y la metodología utilizada para el diseño del sistema.

## 2.1 Fundamentación de los lenguajes de programación

Antes de comenzar es necesario explicar que un lenguaje de programación es un lenguaje escrito utilizado para escribir programas que se ejecutan en un computador. Existen numerosos lenguajes de programación utilizados para el desarrollo de aplicaciones de escritorio, entre los cuales están:

- Pascal
- Delphi
- C++

### 2.1.1 Pascal

El lenguaje de programación en Pascal, es un lenguaje de alto nivel, y de propósito general, lo cual quiere decir que se puede utilizar para cualquier tipo de propósitos. El lenguaje de programación en Pascal se considera un lenguaje estructurado, sencillo y práctico para todos aquellos usuarios que se inician en el mundo de la programación, ya que fue creado con fines de aprendizaje.

Al ser Pascal un lenguaje estructurado, sirve de base para cualquier otro lenguaje de alto nivel, por estas características es utilizado en las universidades e institutos de educación para inicializar a los futuros ingenieros en sistemas o informática.

El lenguaje de programación Pascal, es idóneo en el estudio y definición de las estructuras de datos, su fácil definición lo hace manejable para un programador novato.

Con la programación en Pascal, se pueden realizar desde programas formales, rutinas, utilitarios, hasta cualquier clase de video juegos.

Programación en Pascal es un lenguaje de sintaxis sencilla, muy estructurado y que comprueba exhaustivamente todo tipo de datos.

El mejor de los propósitos de programación en Pascal es que enseña buenas formas de programación, con lo cual se utiliza mucho en la enseñanza, por todos los motivos nominados anteriormente, por su sencillez, su estructuración y su facilidad de lectura y entendimiento. [9]

### 2.1.2 Delphi

Existen diferentes ediciones de programación Delphi, la más básica y económica es la estándar, que incluye los elementos fundamentales de Delphi. La siguiente es la versión profesional, que tiene más componentes, así como códigos fuentes prefabricados; la versión cliente/servidor, posibilita las acciones para internet o intranet. La más completa es la versión empresarial de programación delphi, dedicada al desarrollo de sistemas y aplicaciones complejos. Una de las mayores ventajas de la programación delphi es que es una programación orientada a objeto.

Se conoce como Objeto a los diferentes componentes visuales con que trabaja Delphi, por lo que son objetos una ficha, un botón, una lista, etc. La ventana principal de programación Delphi se denomina Paleta de Componentes. Un programa en Delphi consta de las siguientes secciones:

- **Un compilador:** el cual crea el ejecutable.
- **Una librería:** conocida como VCL, la cual es una librería de clases.
- **El IDE:** Que viene siendo el ambiente de desarrollo integrado (integrated development environment) programación Delphi almacena la aplicación con archivos de extensión .dpr, que define al proyecto en si. Y archivos .dfm el que guarda el formulario creado en Delphi. [10]



### 2.1.3 C++

El lenguaje de programación C++, sin duda el mejor lenguaje de programación actual, a pesar de tener más de 20 años y de que actualmente lenguajes que se supone más efectivos que él intenten arrebatárle un terreno en el cual ha sido el rey durante todo este tiempo. Con C++ podrás programar todo lo que quieras a la máxima velocidad, con una gran facilidad.

C++ agrega a C la posibilidad de trabajar con clases, usar el tipo de dato bool, funciones en línea, sobrecarga de funciones y de operadores, espacios de nombre, argumentos por defecto en las funciones, templates, etc. Lo cual lo define como un lenguaje de programación orientado a objeto. [11]

#### Porque programar en C++?

Una de las razones de programar en C++ es su increíble versatilidad. Con él pueden programarse desde los programas más simples, a los programas más complicados como incluso sistemas operativos. Además es portable, es decir, un programa con el código escrito en C++, se podrá compilar en cualquier sistema operativo o sistema informático si necesidad de cambiar casi el código fuente. Este es por ejemplo uno de los grandes secretos de Linux, al estar el código escrito en este lenguaje(al menos en su concepción original), es más fácil portarlo a diferentes ordenadores como PC's, Macintosh, incluso superordenadores. Además otras de las grandes ventajas del C++, es que es un lenguaje multi-nivel, es decir, puedes usarlo tanto para programar directamente el hardware(dependiendo del sistema operativo, eso si), como para crear aplicaciones tipo Windows definidas todas por poseer una misma interfaz. [12]

Las principales características del C++ son el soporte para programación orientada a objetos y el soporte de plantillas o programación genérica (templates). Se puede decir que C++ es un lenguaje que abarca tres paradigmas de la programación: la programación estructurada, la programación genérica y la programación orientada a objetos.

Además posee una serie de propiedades difíciles de encontrar en otros lenguajes de alto nivel:

- Posibilidad de redefinir los operadores (sobrecarga de operadores).
- Identificación de tipos en tiempo de ejecución (RTTI). [13]

### 2.2 Compilador de C++

#### Borland C++Builder 6 Professional

Borland C++Builder 6 Professional incluye todas las características de la versión Standard junto con un amplio conjunto de herramientas de acceso a bases de datos.

- **Reutilización de componentes** - Ahorre muchas horas de reescritura de código reutilizando los nuevos componentes en futuros proyectos. Los componentes servidor ActiveX y COM, el completo conjunto de componentes Microsoft Office Automation Controller y más de 150 componentes VCL le facilitarán la creación de aplicaciones profesionales en gran medida.
- **Los datos en la red** - Con el nuevo WebBroker y los componentes nativos para Internet ahora puede insertar fácilmente funcionalidades de Internet y TCP/IP (como e-mail, web y grupos de noticias) dentro de los formularios de sus aplicaciones.
- **Las aplicaciones de bases de datos más rápidas** - Los componentes InterBase Express ofrecen a sus aplicaciones un alto rendimiento y bajo mantenimiento, con toda la potencia de InterBase. Alternativamente, puede utilizar controladores para Access97, FoxPro, Paradox y dBASE o la nueva clase TcustomConnection para integrar fácilmente sus propias aplicaciones de bases de datos.
- **Rápida depuración** - La herramienta integrada CodeGuard permite identificar pérdidas en la memoria y recursos. Con el depurador Just-in-Time podrá acceder instantáneamente a los problemas del código fuente durante la ejecución de sus programas.
- **Mayor productividad** - Dispone de otras muchas herramientas para mejorar su productividad, como Advanced Project Manager basado en XML, Control Panel Wizard, Code Completion e InstallShield Express, entre otras. [14]

C++ Builder es una aplicación Windows que proporciona un entorno de trabajo visual para construir aplicaciones Windows que integra distintos aspectos de la programación en un entorno unificado o integrado. La integración y facilidad de manejo hace que sea una herramienta indispensable para el desarrollo rápido de aplicaciones o RAD (Rapid Application Development). Guarda una gran similitud con el IDE de Visual Basic.

C++ Builder trabaja sobre la misma línea que Delphi y Visual Basic. Este desarrollo rápido de aplicaciones hace más fácil la construcción de sus proyectos. Con estas herramientas de programación usted tiene más poder con menos responsabilidad. C++ Builder habilita y produce aplicaciones ejecutables bajo Windows con soporte de 16 y 32 bits. El código se ejecutará mucho más rápido.

C++ Builder necesita una PC con un procesador lo suficientemente poderoso y una amplia memoria. Mientras más poderosa sea su PC, más productiva puede ser. C++ Builder es un producto bastante poderoso y como tal hace significantes demandas en su máquina. La experiencia ha demostrado que el Borland C++ Builder 5.0 corre con buenas prestaciones en un PC con procesador AMD K6-II a 300MHz con 64 Mb de memoria RAM. [15]

### **2.3 Fundamentación de la metodología de desarrollo utilizada**

En la actualidad se tiene como norma que para que un software este bien estructurado, debe seguir una metodología de desarrollo. Una metodología para el desarrollo de un proceso de software es un conjunto de fases, procedimientos, técnicas y reglas; por eso escoger una la metodología que guíe el proceso de desarrollo del sistema es un paso muy importante.

#### **2.3.1 El Proceso Unificado de Modelado (RUP)**

El Proceso Unificado de Rational (RUP, el original inglés Rational Unified Process) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. RUP es en realidad un refinamiento realizado por Rational Software del más genérico Proceso Unificado.

El RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización.

**Por otro lado en lo que se refiere a la metodología esta comprende tres frases claves:**

- Dirigido por los casos de uso.
- Centrado en la arquitectura.
- Iterativo e incremental.

**El RUP está basado en 6 principios clave:**

1. **Adaptar el proceso:** El proceso deberá adaptarse a las características propias del proyecto u organización. El tamaño del mismo, así como su tipo o las regulaciones que lo condicionen, influirán en su diseño específico.
2. **Balancear prioridades:** Los requerimientos de los diversos inversores pueden ser diferentes, contradictorios o disputarse recursos limitados. Debe encontrarse un balance que satisfaga los deseos de todos.
3. **Colaboración entre equipos:** El desarrollo de software no lo hace una única persona sino múltiples equipos. Debe haber una comunicación fluida para coordinar requerimientos, desarrollo, evaluaciones, planes, resultados, etc.
4. **Demostrar valor iterativamente:** Los proyectos se entregan, aunque sea de un modo interno, en etapas iteradas. En cada iteración se analiza la opinión de los inversores, la estabilidad y calidad del producto, y se refina la dirección del proyecto.
5. **Elevar el nivel de abstracción:** Este principio dominante motiva el uso de conceptos reutilizables tales como patrón del software, lenguajes 4GL o esquemas (frameworks) por nombrar algunos. Esto previene a los ingenieros de software ir directamente de los requisitos a la codificación de software a la medida del cliente. Un nivel alto de abstracción también permite discusiones sobre diversos niveles arquitectónicos; estos se pueden acompañar por las representaciones visuales de la arquitectura, por ejemplo con UML.

6. **Enfocarse en la calidad:** El control de calidad no debe realizarse al final de cada iteración, sino en todos los aspectos de la producción.

### Ciclo de vida:

El ciclo de vida RUP es una implementación del Desarrollo en espiral. Fue creado ensamblando los elementos en secuencias semi-ordenadas. El ciclo de vida organiza las tareas en fases e iteraciones.

El RUP divide el proceso de desarrollo en ciclos, teniendo un producto final al final de cada ciclo, cada ciclo se divide en fases que finalizan con un hito donde se debe tomar una decisión importante:

- **Concepción:** se hace un plan de fases, se identifican los principales casos de uso y se identifican los riesgos.
- **Elaboración:** se hace un plan de proyecto, se completan los casos de uso y se eliminan los riesgos.
- **Construcción:** se concentra en la elaboración de un producto totalmente operativo y eficiente y el manual de usuario.
- **Transición:** se implementa el producto en el cliente y se entrena a los usuarios. Como consecuencia de esto suelen surgir nuevos requisitos a ser analizados.

### Principales características:

- Forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo).
- Pretende implementar las mejores prácticas en Ingeniería de Software.
- Desarrollo iterativo.
- Administración de requisitos.
- Uso de arquitectura basada en componentes.
- Control de cambios.
- Modelado visual del software.

- Verificación de la calidad del software. [16]

### 2.3.2 Lenguaje Unificado de Modelado (UML)

Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; aún cuando todavía no es un estándar oficial, está apoyado en gran manera por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.

Es importante remarcar que UML es un "lenguaje" para especificar y no un método o un proceso, se utiliza para definir un sistema de software, para detallar los artefactos en el sistema y para documentar y construir es el lenguaje en el que está descrito el modelo. Se puede aplicar en una gran variedad de formas para soportar una metodología de desarrollo de software (tal como el Proceso Unificado de Rational) pero no especifica en sí mismo qué metodología o proceso usar.

**UML cuenta con varios tipos de diagramas, los cuales muestran diferentes aspectos de las entidades representadas.**

**Diagramas de estructura:** Enfatizan en los elementos que deben existir en el sistema modelado:

- Diagrama de clases.
- Diagrama de componentes.
- Diagrama de objetos.
- Diagrama de estructura compuesta (UML 2.0).
- Diagrama de despliegue.
- Diagrama de paquetes.

**Diagramas de comportamiento:** Enfatizan en lo que debe suceder en el sistema modelado:

- Diagrama de actividades.
- Diagrama de casos de uso.
- Diagrama de estados.

**Diagramas de Interacción:** Un subtipo de diagramas de comportamiento, que enfatiza sobre el flujo de control y de datos entre los elementos del sistema modelado:

- Diagrama de secuencia.
- Diagrama de comunicación.
- Diagrama de tiempos (UML 2.0).
- Diagrama de vista de interacción (UML 2.0). [17]

### 2.4 Tecnologías a emplear

Para el desarrollo de la aplicación se realizó un estudio de las posibles tecnologías a utilizar teniéndose en cuenta la tendencia actual, características y las ventajas de cada una de las tecnologías.

Para la implementación de la aplicación, el lenguaje a utilizar será C++, con el cual pueden programarse desde los programas más simples, hasta los más complicados, como los sistemas operativos. Además es portable, es decir, un programa con el código escrito en C++, se podrá compilar en cualquier sistema operativo o sistema informático sin necesidad de cambiar casi el código fuente.

C++ está basado en la programación orientada a objetos y soporte de plantillas o programación genérica (templates). Se puede decir que C++ es un lenguaje que abarca tres paradigmas de la programación: la programación estructurada, la programación genérica y la programación orientada a objetos.

Para el proceso de desarrollo de software se empleará la metodología RUP, la cual plantea prácticas de desarrollo de software; lo que permite mejores prácticas a utilizar en el modelo de desarrollo de este. Esto se resume en que este proceso de desarrollo está dirigido por casos de uso, centrado en la arquitectura y es iterativo e incremental. Además utiliza UML como lenguaje de modelado el cual permite un mejor entendimiento del proceso de desarrollo de software.

### **Conclusiones**

Con el estudio de las tecnologías y lenguajes que se han abordado en este capítulo se ha llegado a la conclusión de que el sistema se desarrollará utilizando como lenguaje de programación C++, como metodología de desarrollo se utilizará el Proceso Unificado de Desarrollo (RUP) que a su vez hará uso del Lenguaje Unificado de Modelado (UML), para un mejor funcionamiento del sistema.



### Capítulo 3 Características del Sistema

#### Introducción

En este capítulo se analizarán las características que va a tener el sistema, se presentará el diagrama de clases del dominio, así como los requisitos funcionales y no funcionales que debe cumplir el sistema, lo que permitirá tener una visión general del mismo, en cuanto al diagrama de casos de uso del sistema y la descripción de los mismo.

#### 3.1 Modelo del Dominio

Para hacer una descripción de la solución propuesta se analiza el entorno donde se desarrolla el sistema. Existen dos métodos fundamentales para llevarlo a cabo, uno puede ser el Modelo del Negocio, que es una técnica para comprender los procesos de negocio de la organización; y por otra parte está el Modelo del Dominio, que permite la captura de los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan las “cosas” que existen o los eventos que suceden en el entorno en que trabaja el sistema. Teniendo en cuenta que el negocio que se está estudiando, no tiene un negocio bien definido, se propone realizar Modelo del Dominio al sistema en cuestión.

RUP propone para estos casos realizar un modelo del dominio que es una representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés, el cual se representa fundamentalmente por diagramas de clases en UML. Este modelo va a contribuir posteriormente a identificar algunas clases que se utilizarán en el sistema.

El sistema está dirigido a empresas y estudiantes que realizan la programación de la producción, no hay restricción de usuarios en el sistema, por tanto no es posible identificar una estructura o una dinámica organizacional. Por esta razón, no es posible identificar un modelo del negocio y en consecuencia la alternativa es desarrollar el modelo del dominio, el cual se define a continuación.

### 3.2 Diagrama de Clases del Modelo de Dominio

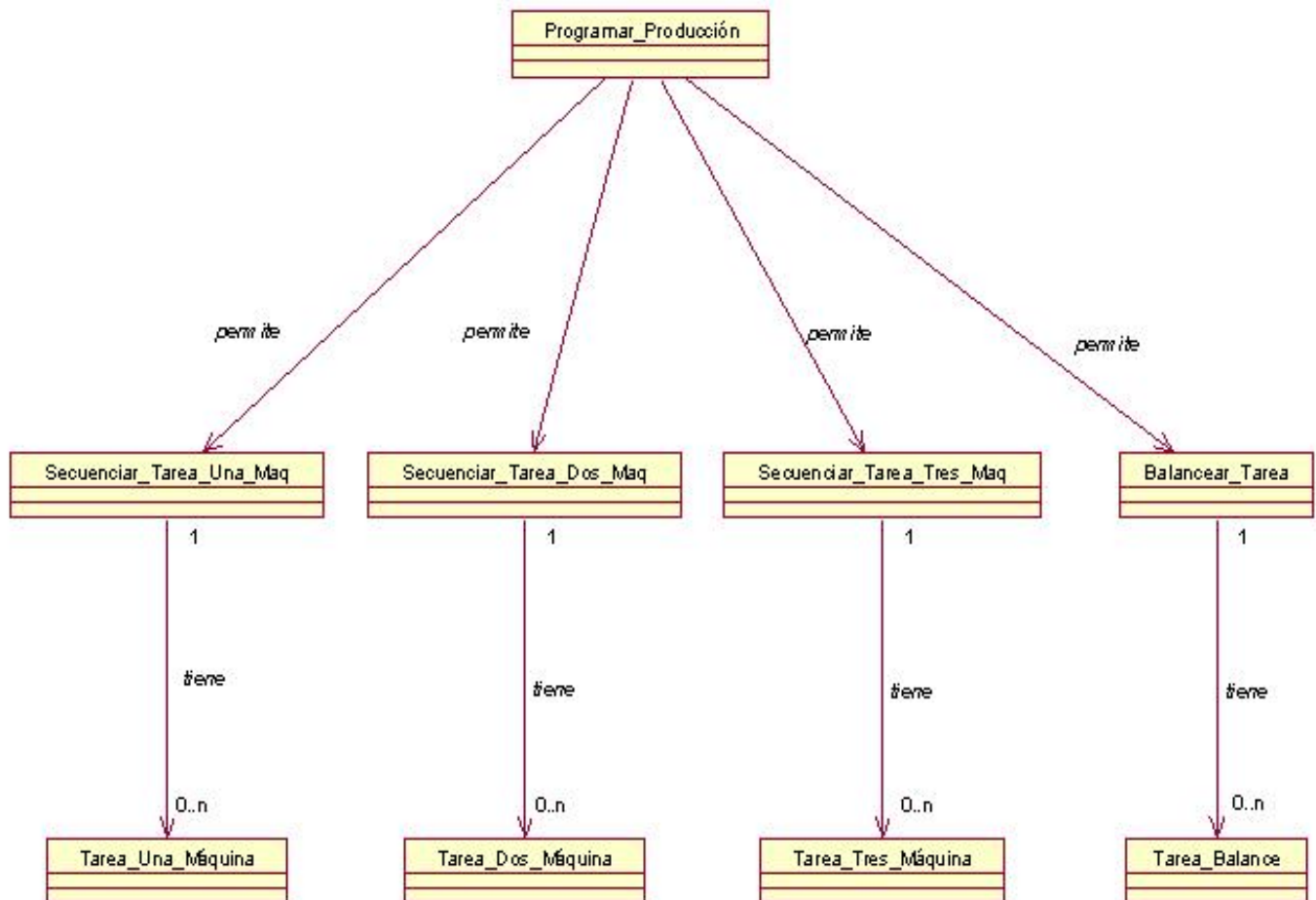


Figura 1 Diagrama de Clases del Dominio

### 3.3 Requerimientos funcionales y no funcionales del sistema

#### 3.3.1 Requerimientos funcionales (RF)

1. Gestionar tareas para una máquina.
2. Crear tarea para una máquina.

3. Seleccionar forma de secuenciar las tareas.
4. Visualizar resultado de secuencia para una máquina.
5. Brindar resumen de las formas de secuenciación.
6. Salvar tareas de una máquina.
7. Gestionar tareas para dos máquinas.
8. Crear tarea para dos máquinas.
9. Secuenciar las tareas para dos máquinas.
10. Visualizar distribución de los tiempos de las tareas por máquinas.
11. Graficar las tareas para dos máquinas.
12. Salvar tareas de dos máquinas.
13. Gestionar tareas para tres máquinas.
14. Crear tarea para tres máquinas.
15. Secuenciar las tareas para tres máquinas.
16. Visualizar distribución de los tiempos de las tareas por máquinas.
17. Graficar las tareas para tres máquinas.
18. Salvar tareas de tres máquinas.
19. Gestionar tareas de balance.
20. Crear tarea de balance.
21. Balancear las tareas.
22. Distribución de las tareas de balance por estaciones de trabajo.
23. Salvar tareas de balance.
24. Cargar tareas
25. Verificar fichero.

### 3.3.2 Requerimientos no Funcionales (RNF).

#### **Apariencia o interfaz externa:**

#### **Usabilidad:**

- El sistema podrá ser usado por cualquier persona que posea conocimientos básicos acerca de la programación de la producción, así como, tener un nivel básico en computación.

#### **Rendimiento:**

- Tiempos de respuestas rápidos al igual que la velocidad de procesamiento de la información.

#### **Político – Culturales:**

- Se debe tener en cuenta un proceso que garantice la integridad de la información, controlando que esté acorde con los valores de nuestra sociedad.

#### **De ayuda y documentación:**

- Debe disponerse de una ayuda bien detallada sobre las principales opciones de manipulación sobre la herramienta.

#### **Software:**

Para que la aplicación pueda ser usada, el ordenador debe cumplir con los siguientes parámetros:

- El sistema se desarrollará en C++ Builder.
- Sistema Operativo Windows 2000 como mínimo.

#### **Hardware:**

- Como mínimo 64MB de RAM y espacio libre en disco duro de acuerdo al uso de la herramienta.

### 3.4 Diagrama de Caso de Uso del Sistema

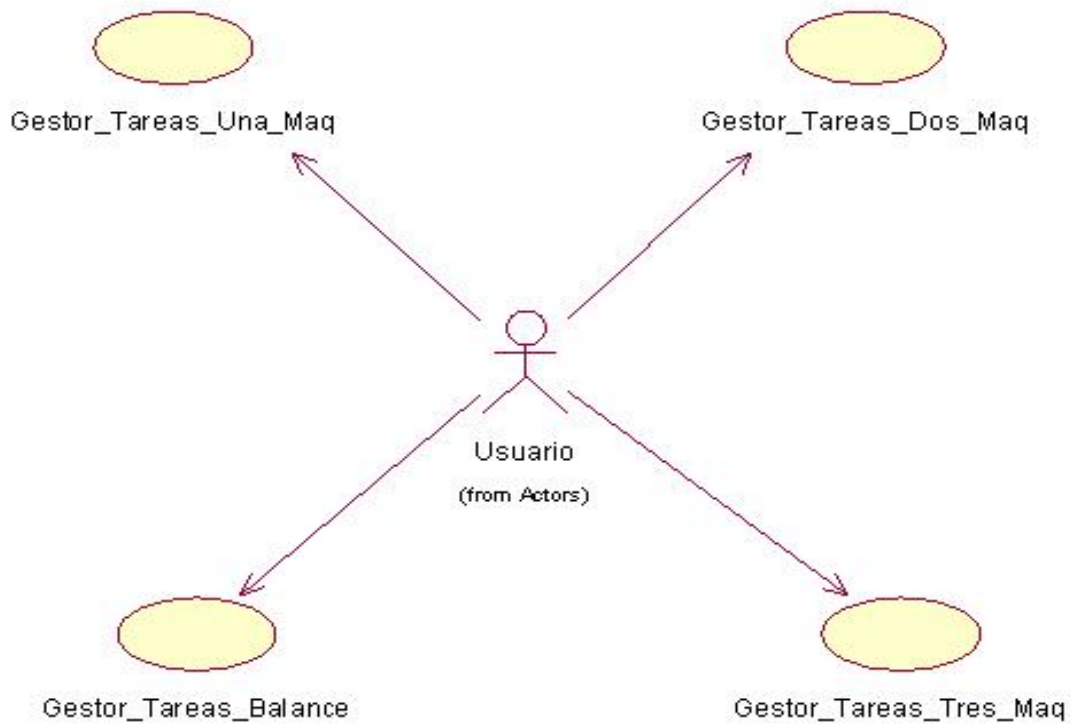


Figura 2 Diagrama de CU del Sistema

### 3.5 Descripción de los actores y diagrama de los casos de uso del sistema a automatizar

#### 3.5.1 Actores del sistema y su justificación

##### Justificación de los actores del sistema

Actores del Sistema	del	Justificación
Usuario		Es el encargado de realizar todas las operaciones del sistema.

Tabla 1 Justificación de los actores del sistema

3.5.2 Descripción de los Casos de Uso del Sistema.

Descripción del CU Gestionar tareas para una máquina

<b>Nombre del CU</b>	Gestionar tareas para una máquina
<b>Actor</b>	Usuario(Inicia)
<b>Propósito</b>	Gestionar las tareas para una máquina.
<b>Descripción:</b> El Caso de Uso se inicia cuando el usuario abre la aplicación y despliega el menú y selecciona la opción de secuenciar tareas para una máquina, luego introduce la cantidad de tareas y pulsa el botón insertar número de tareas a secuenciar, después introduce los datos correspondientes en la interfaz y selecciona la regla por la que desea secuenciar, lo cual le muestra una interfaz donde aparecen la secuencia de las tareas y otra información. Además la interfaz permite dar un resumen de las reglas de secuenciación, dando click en el menú de esa misma interfaz.	
<b>Referencia</b>	RF 1, 2, 3, 4, 5, 6, 24, 25
<b>Precondiciones</b>	El usuario debe pasar los datos necesarios para realizar dicho caso de uso.
<b>Poscondiciones</b>	Quedan secuenciadas las tareas.
<b>Curso Normal de los Eventos</b>	
<b>Acciones del actor</b>	<b>Respuesta del Sistema</b>
1. El usuario selecciona la opción de secuenciar tareas en una máquina.	1.1. El sistema muestra la interfaz donde le pide la cantidad de tareas a secuenciar.
2. El usuario teclea la cantidad de tareas y hace click en el botón insertar.	2.1. El sistema en la misma interfaz permite pasar los datos correspondientes a las tareas.
3. El usuario introduce los datos de las tareas.	3.1 El sistema crea las tareas con los datos introducidos.

4. El usuario selecciona la regla de secuenciación.	4.1. El sistema muestra otra interfaz donde aparecen los resultados de la regla de secuenciación y una serie de información.
5. El usuario selecciona la opción de ver los resúmenes de los resultados de las reglas de secuenciación.	5.1 El sistema brinda la información solicitada.
6. El usuario selecciona la opción salvar del menú de la interfaz donde el introduce los datos de las tareas.	6.1 El sistema permite salvar las tareas en un fichero.
7. El usuario selecciona la opción cargar del menú que esta en dicha interfaz.	7.1 El sistema muestra una ventana donde le permitirá buscar el fichero que desea cargar.
8. El usuario selecciona el fichero que desea cargar y presiona click en abrir.	8.1 El sistema verifica el identificador del fichero seleccionado. 8.2 El sistema carga el fichero.
<b>Curso Alterno</b>	
2. Si el usuario no introduce la cantidad de tarea.	2.1 El sistema muestra un mensaje que informa que debe insertar la cantidad de tarea.
3. Si el usuario no introduce los datos de las tareas.	3.1 El sistema muestra un mensaje que informa que hay que introducir los datos de las tareas.
	8.1 Si el identificador del fichero seleccionado no es el correcto, el sistema muestra un mensaje de error. 8.2 Se repite el flujo normal desde el paso 7 de las acciones del estudiante.
<b>Prioridad</b>	Crítico

Tabla 2 Descripción del CU Gestionar tareas para una máquina

**Descripción del CU Gestionar tareas para dos máquina**

<b>Nombre del CU</b>	Gestionar tareas para dos máquina
<b>Actor</b>	Usuario (Inicia)
<b>Propósito</b>	Gestionar las tareas para dos máquinas.
<b>Descripción:</b> El Caso de Uso se inicia cuando el usuario abre la aplicación y despliega el menú y selecciona la opción de secuenciar tareas en dos máquina, luego introduce la cantidad de tareas y pulsa el botón insertar, después introduce los datos correspondientes en la interfaz y pulsa el botón secuenciar, luego le muestra una interfaz donde aparecen los resultados deseados; y por último pulsa el botón graficar el cual le muestra una gráfica con una información de utilidad para el estudiante.	
<b>Referencia</b>	RF 7, 8, 9, 10, 11,12, 24, 25
<b>Precondiciones</b>	El usuario debe pasar los datos necesarios para realizar dicho caso de uso.
<b>Poscondiciones</b>	Quedan secuenciadas las tareas.
<b>Curso Normal de los Eventos</b>	
<b>Acciones del actor</b>	<b>Respuesta del Sistema</b>
1. El usuario selecciona la opción de secuenciar tareas en dos máquinas.	1.1. El sistema muestra la interfaz donde le pide la cantidad de tareas a secuenciar.
2. El usuario teclea la cantidad de tareas y hace click en el botón insertar.	2.1. El sistema en la misma interfaz permite pasar los datos correspondientes a las tareas.
3. El usuario introduce los datos de las tareas.	3.1 El sistema crea las tareas con los datos introducidos.
4. El usuario pulsa el botón secuenciar.	4.1. El sistema muestra otra interfaz donde aparecen los resultados de la secuenciación y una serie de información.



5. El usuario pulsa el botón graficar.	5.1. El sistema muestra otra interfaz en la que permite ver la gráfica con información necesaria para el estudiante.
6. El usuario selecciona la opción salvar del menú de la interfaz donde él introduce los datos de las tareas.	6.1 El sistema permite salvar las tareas en un fichero.
7.-El usuario selecciona la opción cargar del menú que esta en dicha interfaz.	7.1 El sistema muestra una ventana donde le permitirá buscar el fichero que desea cargar.
8.- El usuario selecciona el fichero que desea cargar y presiona click en abrir.	8.1 El sistema verifica el identificador del fichero seleccionado. 8.2 El sistema carga el fichero.
<b>Curso Alterno</b>	
2. Si el usuario no introduce la cantidad de tareas.	2.1 El sistema muestra un mensaje que informa que debe insertar la cantidad de tareas.
3. Si el usuario no introduce los datos de las tareas.	3.1. Si los datos de las tareas no son introducidos, el sistema muestra un mensaje que informa que hay que introducir estos.
	8.1 Si el identificador del fichero seleccionado no es el correcto, el sistema muestra un mensaje de error. 8.2 Se repite el flujo normal desde el paso 7 de las acciones del estudiante.
<b>Prioridad</b>	Crítico

Tabla 3 Descripción del CU Gestionar tareas para dos máquina

**Descripción del CU Gestionar tareas para tres máquina**

<b>Nombre del CU</b>	Gestionar tareas para tres máquina
<b>Actor</b>	Usuario (Inicia)
<b>Propósito</b>	Gestionar las tareas para tres máquinas.
<b>Descripción:</b> El Caso de Uso se inicia cuando el usuario abre la aplicación y despliega el menú y selecciona la opción de secuenciar tareas en dos máquinas, luego introduce la cantidad de tareas y pulsa el botón insertar, después introduce los datos correspondientes en la interfaz y pulsa el botón secuenciar, luego le muestra una interfaz donde aparecen los resultados deseados; y por último pulsa el botón graficar el cual le muestra una gráfica con una información de utilidad para el estudiante.	
<b>Referencia</b>	RF 13, 14, 15, 16, 17,18, 24, 25
<b>Precondiciones</b>	El usuario debe pasar los datos necesarios para realizar dicho caso de uso.
<b>Poscondiciones</b>	Queda secuenciadas las tareas.
<b>Curso Normal de los Eventos</b>	
<b>Acciones del actor</b>	<b>Respuesta del Sistema</b>
1. El usuario selecciona la opción de secuenciar tareas en tres máquinas.	1.1. El sistema muestra la interfaz donde le pide la cantidad de tareas a secuenciar.
2. El usuario teclea la cantidad de tareas y hace click en el botón insertar.	2.1. El sistema en la misma interfaz permite pasar los datos correspondientes a las tareas.
3. El usuario introduce los datos de las tareas.	3.1 El sistema crea las tareas con los datos introducidos.
4. El usuario pulsa el botón secuenciar.	4.1. El sistema muestra otra interfaz donde aparecen los resultados de la secuenciación y una serie de información.

5. El usuario pulsa el botón graficar.	5.1. El sistema muestra otra interfaz en la que permite ver la gráfica con información necesaria para el estudiante.
6. El usuario selecciona la opción salvar del menú de la interfaz donde el introduce los datos de las tareas.	6.1 El sistema permite salvar las tareas en un fichero.
7.-El usuario selecciona la opción cargar del menú que esta en dicha interfaz.	7.1 El sistema muestra una ventana donde le permitirá buscar el fichero que desea cargar.
8.- El usuario selecciona el fichero que desea cargar y presiona click en abrir.	8.1 El sistema verifica el identificador del fichero seleccionado. 8.2 El sistema carga el fichero.
<b>Curso Alterno</b>	
2. Si el usuario no introduce la cantidad de tarea.	2.1 El sistema muestra un mensaje que informa que debe insertar la cantidad de tarea.
3. Si el usuario no introduce los datos de las tareas.	3.1. Si los datos de las tareas no son introducidos, el sistema muestra un mensaje que informa que hay que introducir estos.
	8.1 Si el identificador del fichero seleccionado no es el correcto, el sistema muestra un mensaje de error. 8.2 Se repite el flujo normal desde el paso 7 de las acciones del estudiante.
<b>Prioridad</b>	Crítico

Tabla 4 Descripción del CU Gestionar tareas para tres máquina

**Descripción del CU Gestionar tareas de balance**

<b>Nombre del CU</b>	Gestionar tareas de balance
<b>Actor</b>	Usuario (Inicia)
<b>Propósito</b>	Gestionar las tareas y distribuirlas por estaciones de trabajo.
<b>Descripción:</b> El Caso de Uso se inicia cuando el usuario abre la aplicación y despliega el menú y selecciona la opción de balancear tareas en dos máquina, luego introduce la cantidad de tareas y el tiempo de ciclo de todas las tareas, luego pulsa el botón insertar, después introduce los datos correspondientes en la interfaz y pulsa el botón balancear, luego le muestra una interfaz donde aparecen los resultados deseados.	
<b>Referencia</b>	RF 19, 20, 21, 22, 23, 24, 25
<b>Precondiciones</b>	El usuario debe pasar los datos necesarios para realizar dicho caso de uso.
<b>Poscondiciones</b>	Queda balanceadas las tareas.
<b>Curso Normal de los Eventos</b>	
<b>Acciones del actor</b>	<b>Respuesta del Sistema</b>
1. El usuario selecciona la opción de balancear tareas.	1.1. El sistema muestra la interfaz donde le pide la cantidad de tareas y el tiempo de ciclo de todas las tareas.
2. El usuario teclea la cantidad de tareas luego hace click en el botón insertar.	2.1. El sistema en la misma interfaz permite pasar los datos correspondientes a las tareas.
3. El usuario introduce los datos de las tareas.	3.1 El sistema crea las tareas con los datos introducidos.
4. El usuario pulsa el botón balancear.	4.1. El sistema muestra otra interfaz donde aparecen los resultados del balance.

5.- El estudiante selecciona la opción salvar del menú de la interfaz donde el introduce los datos de las tareas.	5.1 El sistema permite salvar las tareas en un fichero.
6.-El usuario selecciona la opción cargar del menú que esta en dicha interfaz.	6.1 El sistema muestra una ventana donde le permitirá buscar el fichero que desea cargar.
7.- El usuario selecciona el fichero que desea cargar y presiona click en abrir.	7.1 El sistema verifica el identificador del fichero seleccionado.  7.2 El sistema carga el fichero.
<b>Curso Alterno</b>	
2. Si el usuario no introduce la cantidad de tarea.	2.1 El sistema muestra un mensaje que informa que debe insertar la cantidad de tarea.
3. Si el usuario no introduce los datos de las tareas.	3.1. Si los datos de las tareas no son introducidos, el sistema muestra un mensaje que informa que hay que introducir estos.
	7.1 Si el identificador del fichero seleccionado no es el correcto, el sistema muestra un mensaje de error.  7.2 Se repite el flujo normal desde el paso 7 de las acciones del estudiante.
<b>Prioridad</b>	Crítico

Tabla 5 Descripción del CU Gestionar tareas de balance

### Conclusiones

En este capítulo se ha realizado un estudio detallado sobre la modelación del negocio, el cual fue restringido a un modelo del dominio por no tenerse claro el funcionamiento del negocio de dicho sistema, además se explicó el funcionamiento del sistema, estableciendo los requerimientos funcionales y no funcionales para comprender mucho mejor el problema que la aplicación pretende solucionar. Se presentó el diagrama de casos de uso del sistema y la descripción de cada uno de los casos de uso, para una mejor comprensión de lo que debe hacer el sistema, por lo que ya se puede pasar al diseño e implementación del sistema.

## Capítulo 4 Construcción de la solución propuesta

### Introducción

En este capítulo se realiza el diseño del sistema mediante los diagramas de clases del diseño y los diagramas de secuencias de la realización de los casos de uso de dicho sistema. También se realizará la implementación del sistema mediante la creación de componentes físicos que se traducen en ficheros .h y .cpp del lenguaje C++.

### 4.1 Diagramas de clases del diseño

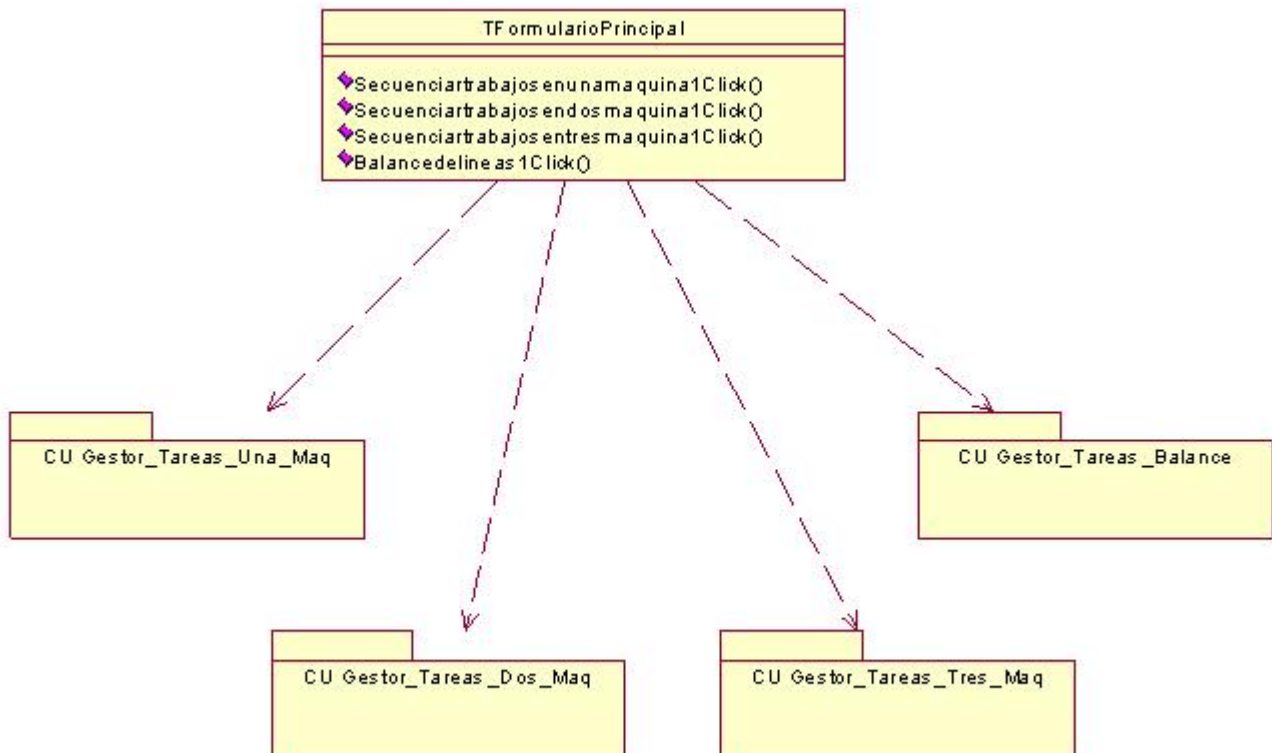


Figura 3 Diagrama de clases del diseño

Diagrama CU Gestionar Tareas Una Máquina

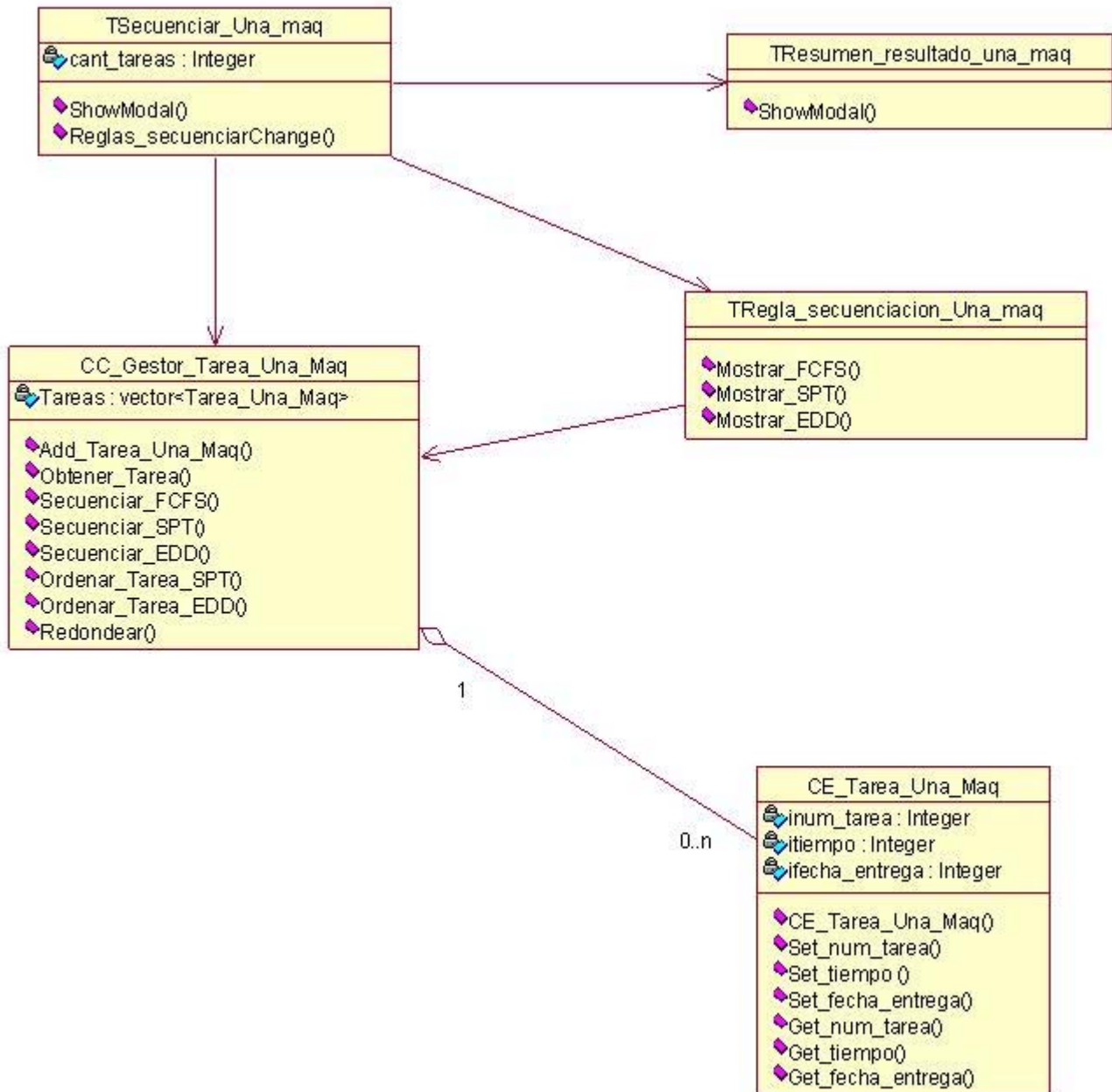


Figura 4 Diagrama CU Gestionar Tareas Una Máquina



Diagrama CU Gestionar Tareas Dos Máquinas

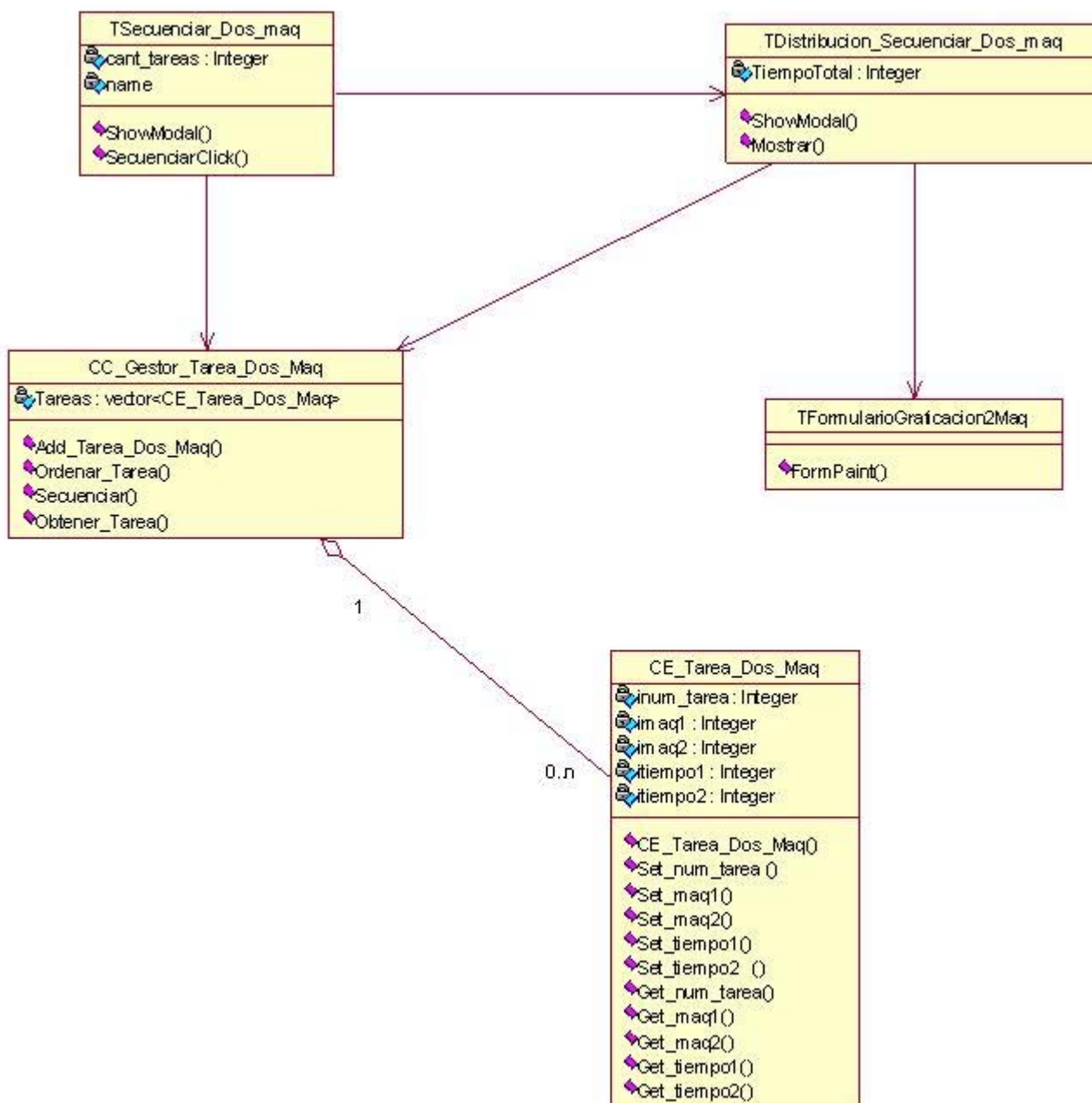


Figura 5 Diagrama CU Gestionar Tareas Dos Máquinas

Diagrama CU Gestionar Tareas Tres Máquinas

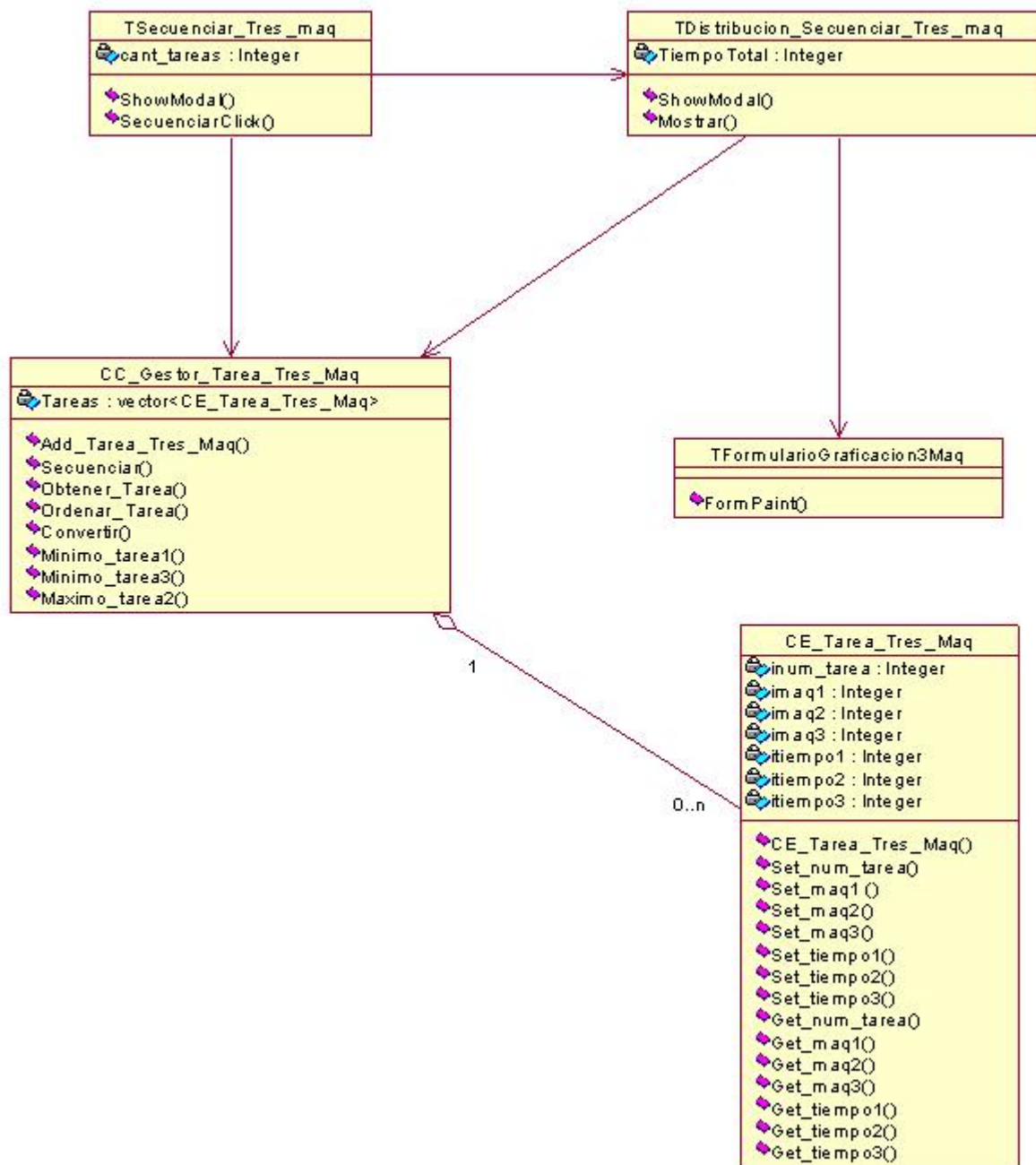


Figura 6 Diagrama CU Gestionar Tareas Tres Máquinas

Diagrama CU Gestionar tareas de balance

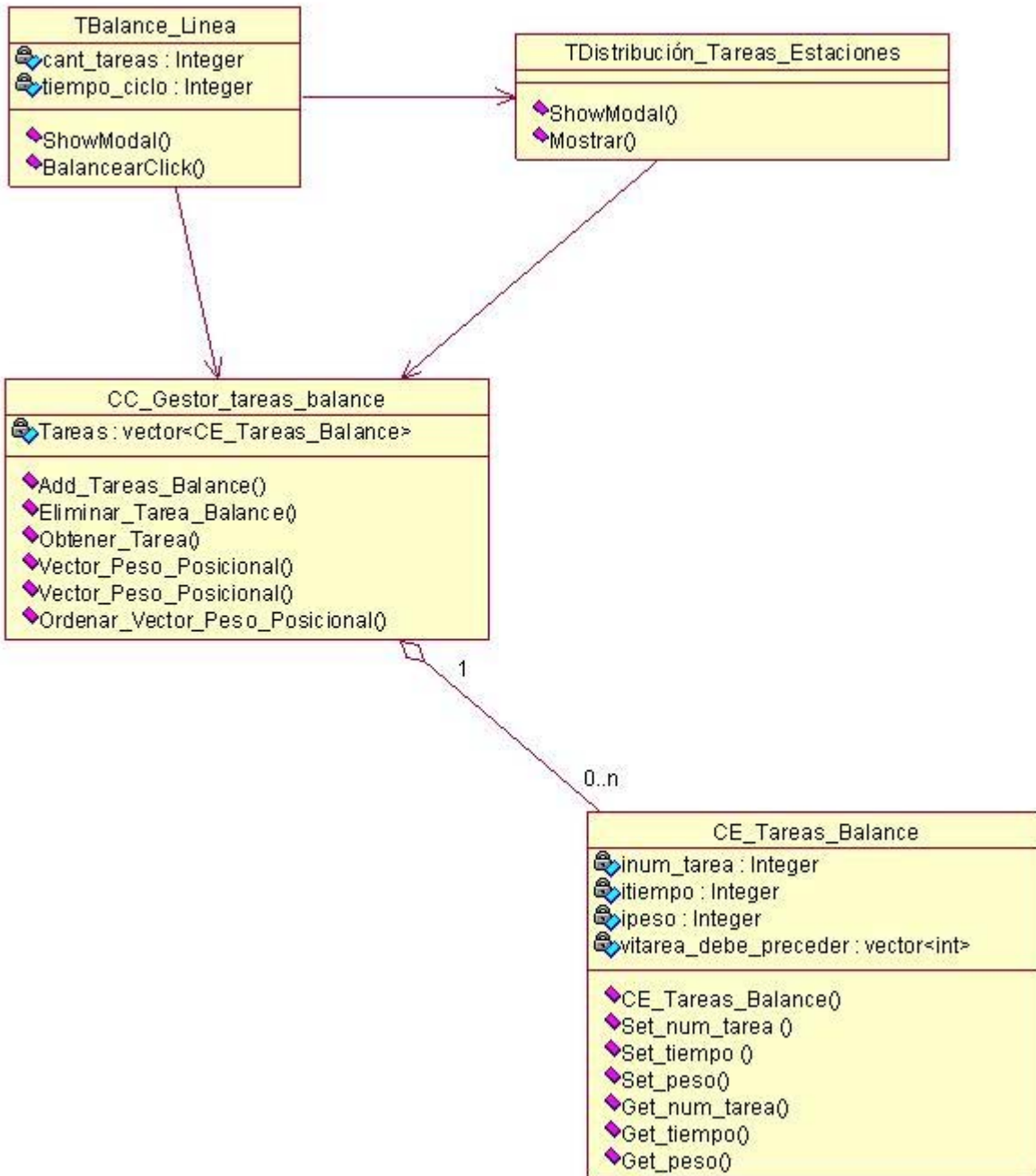


Figura 7 Diagrama CU Gestionar tareas de balance

## 4.2 Diagramas de Interacción

A continuación se presentan los diagramas de secuencias que responden a los casos de uso del sistema.

### Diagrama de Secuencia del CU Gestionar Tareas Una Máquina

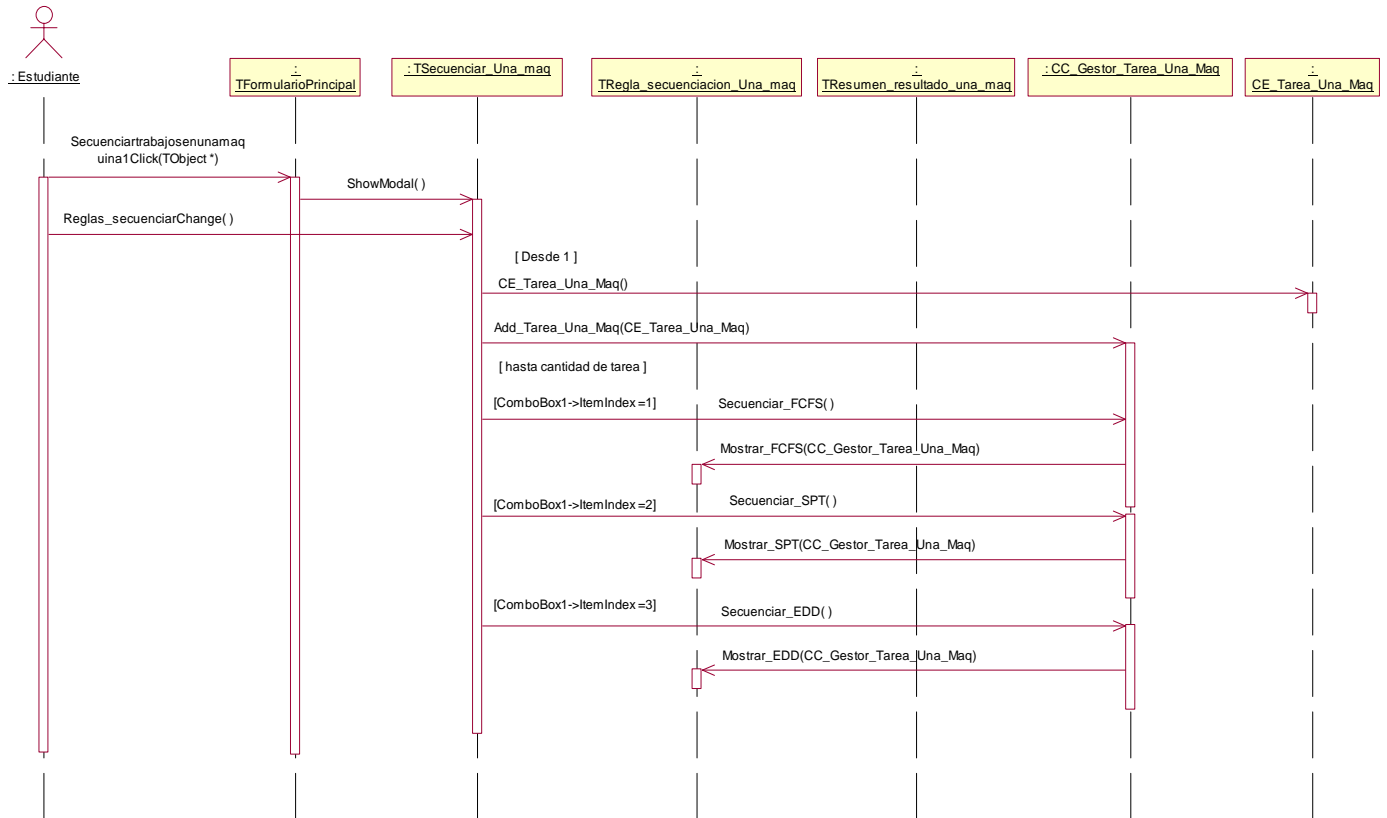


Figura 8 Diagrama de Secuencia del CU Gestionar Tareas Una Máquina

Diagrama de Secuencia del CU Gestionar Tareas Dos Máquinas

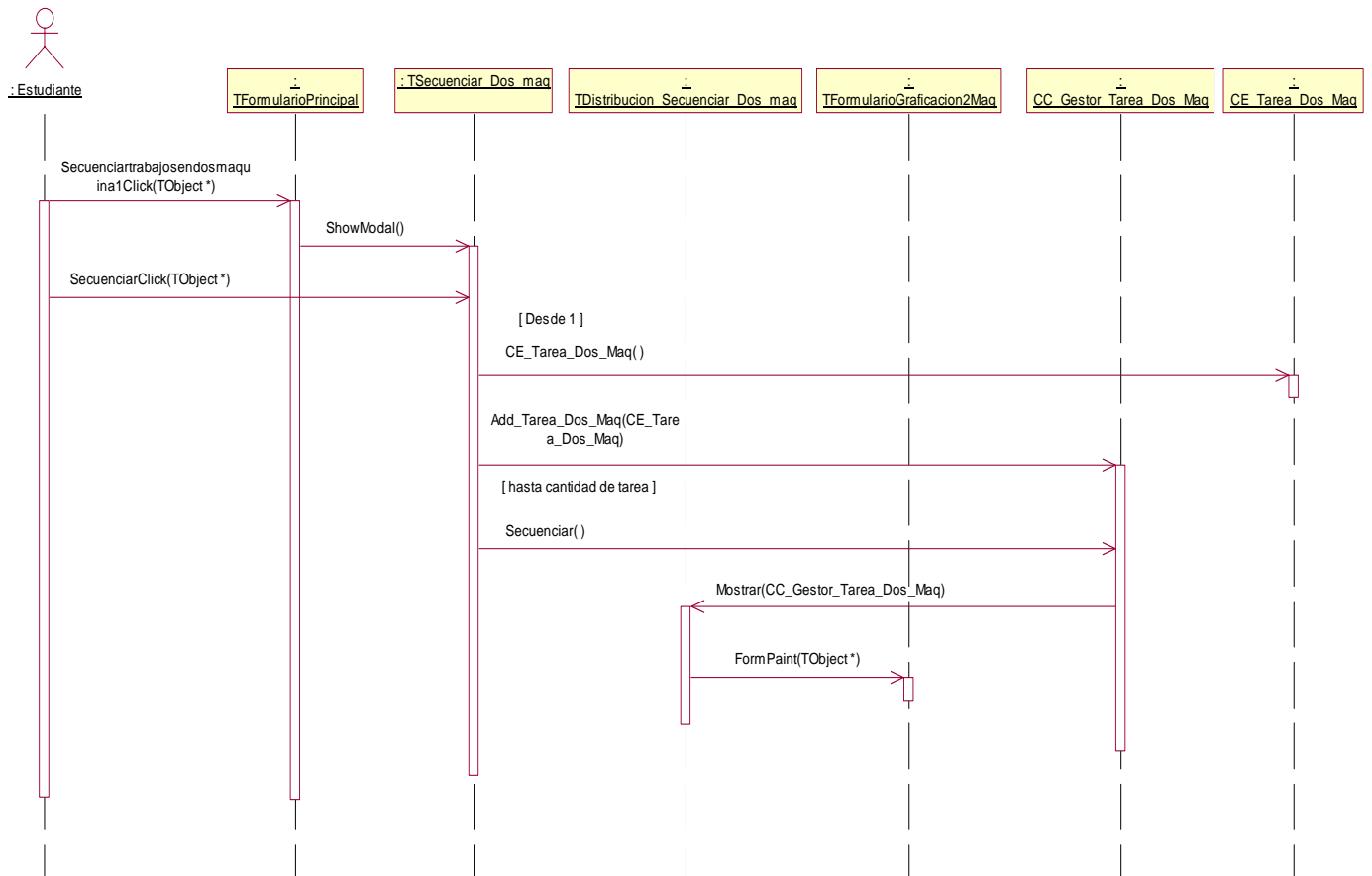


Figura 9 Diagrama de Secuencia del CU Gestionar Tareas Dos Máquinas

Diagrama de Secuencia del CU Gestionar Tareas Tres Máquinas

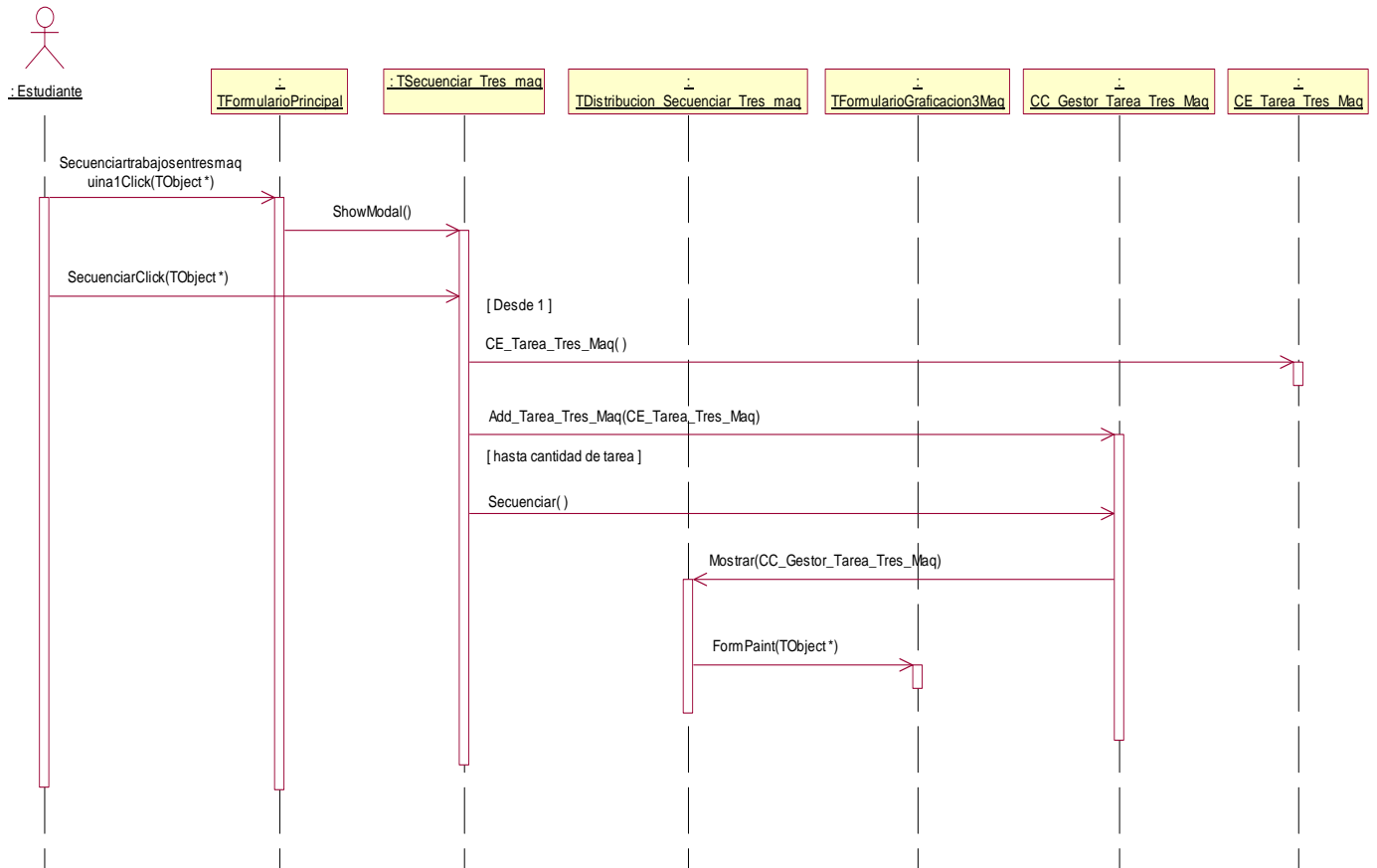


Figura 10 Diagrama de Secuencia del CU Gestionar Tareas Tres Máquinas

Diagrama de Secuencia del CU Gestionar Tareas de Balance

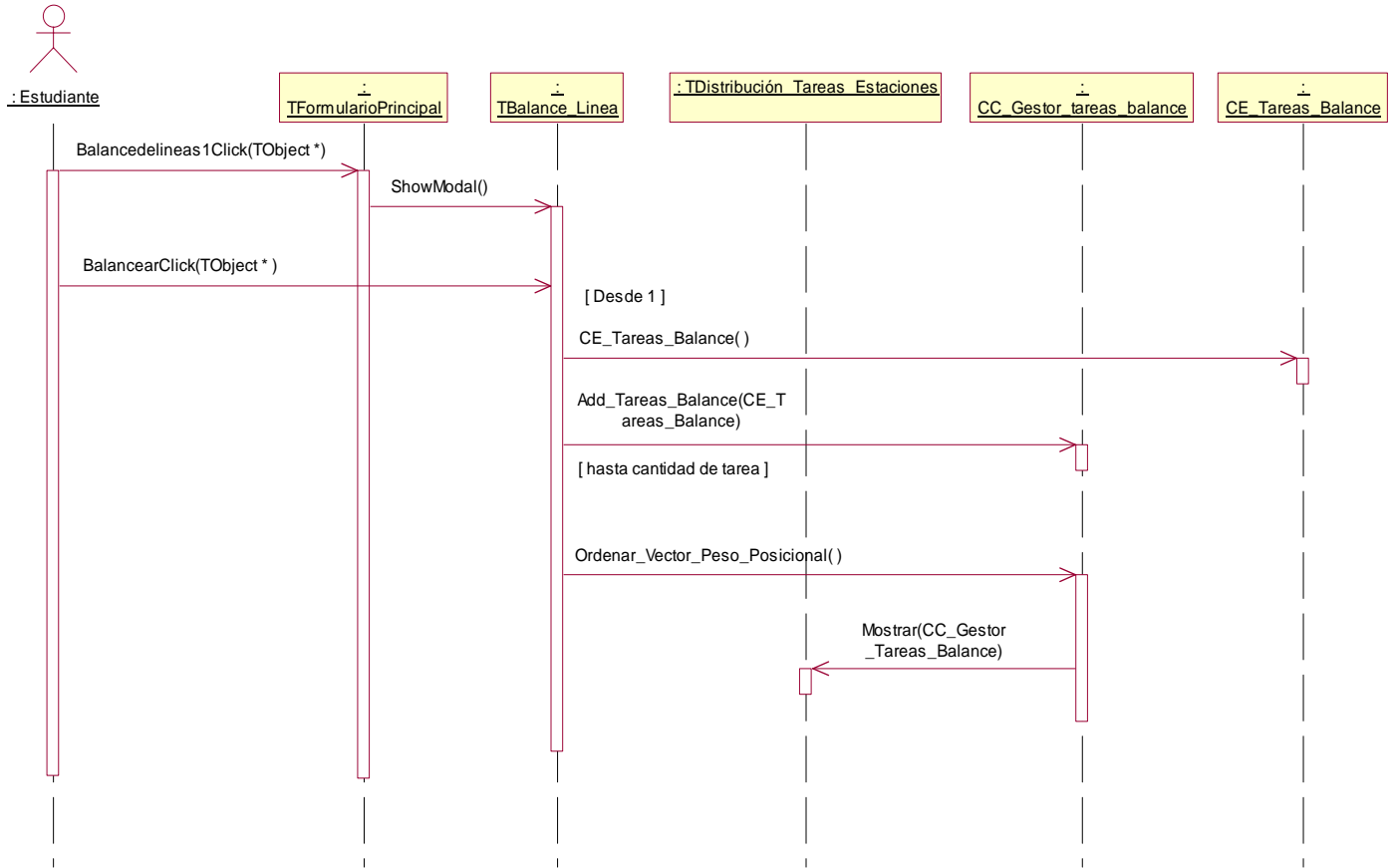


Figura 11 Diagrama de Secuencia del CU Gestionar Tareas de Balance

4.3 Descripción de las clases del diseño

CE\_Tareas\_Balance

<b>Nombre:</b> CE_Tareas_Balance	
<b>Tipo de clase:</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>
inum_tarea	Int
it tiempo	Int
ipeso	Int
vitarea_debe_preceder	Vector<int>

Tabla 6 CE\_Tareas\_Balance

**CE\_Tarea\_Una\_Maq**

<b>Nombre:</b> CE_Tarea_Una_Maq	
<b>Tipo de clase:</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>
inum_tarea	Int
it tiempo	Int
ifecha_entrega	Int

Tabla 7 CE\_Tarea\_Una\_Maq

**Table1 CE\_Tarea\_Dos\_Maq**

<b>Nombre:</b> CE_Tarea_Dos_Maq	
<b>Tipo de clase:</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>
inum_tarea	Int
imaq1	Int
Imaq2	int
it tiempo1	int
It tiempo2	int

Tabla 8 CE\_Tarea\_Dos\_Maq

**CE\_Tarea\_Tres\_Maq**

<b>Nombre:</b> CE_Tarea_Tres_Maq	
<b>Tipo de clase:</b> Entidad	
<b>Atributo</b>	<b>Tipo</b>
inum_tarea	Int
imaq1	Int
Imaq2	int
Imaq3	int
it tiempo1	int
It tiempo2	int
It tiempo3	int

Tabla 9 CE\_Tarea\_Tres\_Maq

**CC\_Gestor\_Tareas\_Balance**

<b>Nombre:</b> CC_Gestor_Tareas_Balance
---



<b>Tipo de clase:</b> Control	
<b>Atributo</b>	<b>Tipo</b>
voTareas	vector<CE_Tareas_Balance>
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	<b>Add_Tareas_Balance</b> (CE_Tareas_Balance otarea)
<b>Descripción:</b>	Adiciona una tarea de balance
<b>Nombre:</b>	<b>Eliminar_Tareas_Balance</b> (int iapos)
<b>Descripción:</b>	Elimina una tarea de balance en una posición dada
<b>Nombre:</b>	<b>Obtener_Tarea</b> (int iapos)
<b>Descripción:</b>	Obtiene una tarea en una posición dada
<b>Nombre:</b>	<b>Vector_Peso_Posicional</b> ()
<b>Descripción:</b>	Obtiene una lista de pesos posicionales
<b>Nombre:</b>	<b>Ordenar_Vector_Peso_Posicional</b> ()
<b>Descripción:</b>	Ordena la lista de pesos posicionales.

Tabla 10 CC\_Gestor\_Tareas\_Balance

### CC\_Gestor\_Tarea\_Una\_Maq

<b>Nombre:</b> CC_Gestor_Tarea_Una_Maq	
<b>Tipo de clase:</b> Control	
<b>Atributo</b>	<b>Tipo</b>
voTareas	vector< CE_Tarea_Una_Maq>
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	<b>Add_Tareas_Balance</b> (CE_Tarea_Una_Maq)
<b>Descripción:</b>	Adiciona una tarea
<b>Nombre:</b>	<b>Obtener_Tarea</b> (int iapos)
<b>Descripción:</b>	Obtiene una tarea en una posición dada
<b>Nombre:</b>	<b>Secuenciar_FCFS</b> ()
<b>Descripción:</b>	Halla la secuencia del conjunto de tareas según el orden de llegada, la primera que llega primero en servirse
<b>Nombre:</b>	<b>Secuenciar_SPT</b> ()
<b>Descripción:</b>	Halla la secuencia del conjunto de tareas según el menor tiempo de procesamiento.
<b>Nombre:</b>	<b>Secuenciar_EDD</b> ()
<b>Descripción:</b>	Halla la secuencia del conjunto de tareas según la menor fecha de entrega
<b>Nombre:</b>	<b>Ordenar_Tarea_SPT</b> ()
<b>Descripción:</b>	Ordena las tareas según el tiempo de procesamiento.
<b>Nombre:</b>	<b>Ordenar_Tarea_SPT</b> ()
<b>Descripción:</b>	Ordena las tareas según la fecha de entrega
<b>Nombre:</b>	<b>Redondear</b> (double dnumero)
<b>Descripción:</b>	Redondea un numero decimal a dos cifras significativas después de la coma

Tabla 11 CC\_Gestor\_Tarea\_Una\_Maq

**CC\_Gestor\_Tarea\_Dos\_Maq**

<b>Nombre:</b> CC_Gestor_Tarea_Dos_Maq	
<b>Tipo de clase:</b> Control	
<b>Atributo</b>	<b>Tipo</b>
voTareas	vector< CE_Tarea_Dos_Maq>
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	<b>Add_Tareas_Balance( CE_Tarea_Dos_Maq)</b>
<b>Descripción:</b>	Adiciona una tarea
<b>Nombre:</b>	<b>Obtener_Tarea(int iapos)</b>
<b>Descripción:</b>	Obtiene una tarea en una posición dada
<b>Nombre:</b>	<b>Secuenciar ()</b>
<b>Descripción:</b>	Halla la secuencia del conjunto de tareas
<b>Nombre:</b>	<b>Ordenar_Tareas()</b>
<b>Descripción:</b>	Ordena las tareas por el tiempo de procesamiento

Tabla 12 CC\_Gestor\_Tarea\_Dos\_Maq

**CC\_Gestor\_Tarea\_Tres\_Maq**

<b>Nombre:</b> CC_Gestor_Tarea_Tres_Maq	
<b>Tipo de clase:</b> Control	
<b>Atributo</b>	<b>Tipo</b>
voTareas	vector< CE_Tarea_Tres_Maq>
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	<b>Add_Tareas_Balance( CE_Tarea_Dos_Maq)</b>
<b>Descripción:</b>	Adiciona una tarea
<b>Nombre:</b>	<b>Obtener_Tarea(int iapos)</b>
<b>Descripción:</b>	Obtiene una tarea en una posición dada
<b>Nombre:</b>	<b>Secuenciar ()</b>
<b>Descripción:</b>	Halla la secuencia del conjunto de tareas
<b>Nombre:</b>	<b>Ordenar_Tareas()</b>
<b>Descripción:</b>	Ordena las tareas por el tiempo de procesamiento
<b>Nombre:</b>	<b>Convertir()</b>
<b>Descripción:</b>	Convierte el problema de tres máquinas para dos máquinas
<b>Nombre:</b>	<b>Mínimo_tarea1()</b>
<b>Descripción:</b>	Obtiene el menor tiempo de procesamiento de las tareas de la primera máquina.
<b>Nombre:</b>	<b>Mínimo_tarea3()</b>
<b>Descripción:</b>	Obtiene el menor tiempo de procesamiento de las tareas de la tercera máquina.
<b>Nombre:</b>	<b>Máximo_tarea2()</b>
<b>Descripción:</b>	Obtiene el mayor tiempo de procesamiento de las tareas de la segunda máquina.

Tabla 13 CC\_Gestor\_Tarea\_Tres\_Maq

### 4.4 Estándares de codificación

Los estándares de codificación que a continuación se definen tienen como objetivos aumentar la portabilidad, reducir el esfuerzo de mantenimiento, y sobre todo, mejorar la legibilidad del código desarrollado en C++.

Se nombrarán los ficheros .h y .cpp de la siguiente forma:

Nombre\_de\_la\_Unit.cpp

#### **Clases:**

Las clases entidad se nombrarán siguiendo el siguiente patrón:

CE\_Nombre\_de\_la\_clase

Las clases controladoras se nombrarán siguiendo el siguiente patrón:

CC\_Nombre\_de\_la\_clase

#### **Variables:**

Los nombres de las variables comenzarán con un identificador del tipo de dato al que correspondan, como se muestra a continuación. En caso de ser argumentos de algún método, se les antepondrá letra “a”.

```
/* tipos simples */
```

```
bool bVarName;
```

```
int iName;
```

```
unsigned int uiName;
```

```
float fName;
```

```
char cName;
```

```
double** dPName; // puntero bidimensional
```

```
vector<tipo> vtipoName
```

#### 4.5 Apariencia de la interfaz y tratamiento de errores

La apariencia de la aplicación es muy importante, puesto que es el medio en el cual el usuario interactúa con el sistema, por lo que debe ser lo más amigable posible.

#### Ejemplo de interfaz: “Ventana principal”

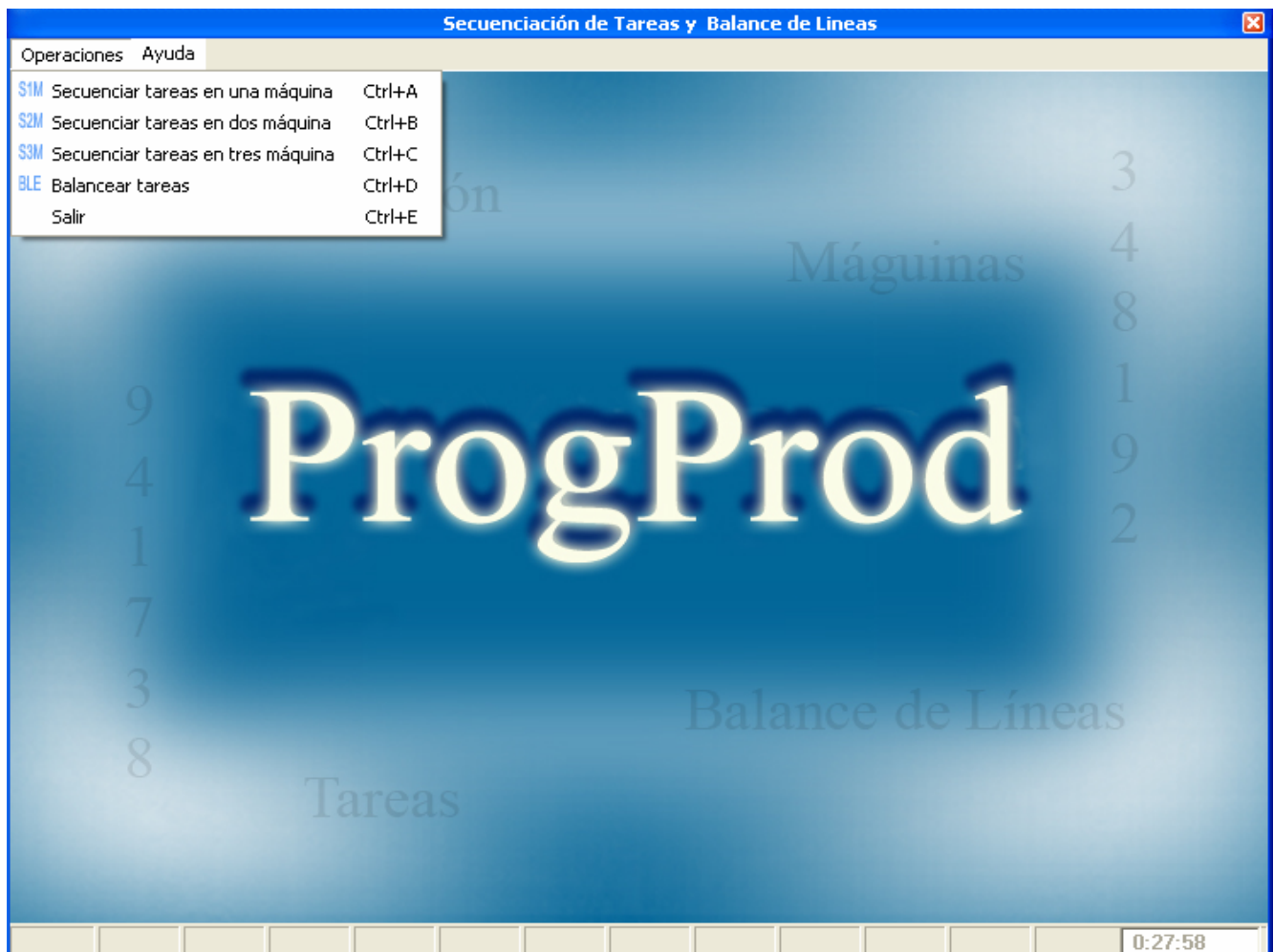


Figura 12 Ejemplo de interfaz: “Ventana principal”

Ejemplo de interfaz: “Ventana: Secuenciar tareas en tres máquinas”



Figura 13 Ejemplo de interfaz: “Secuenciar tareas en tres máquinas”

Ejemplo de interfaz: “Ventana: Graficación de tareas en tres máquinas”

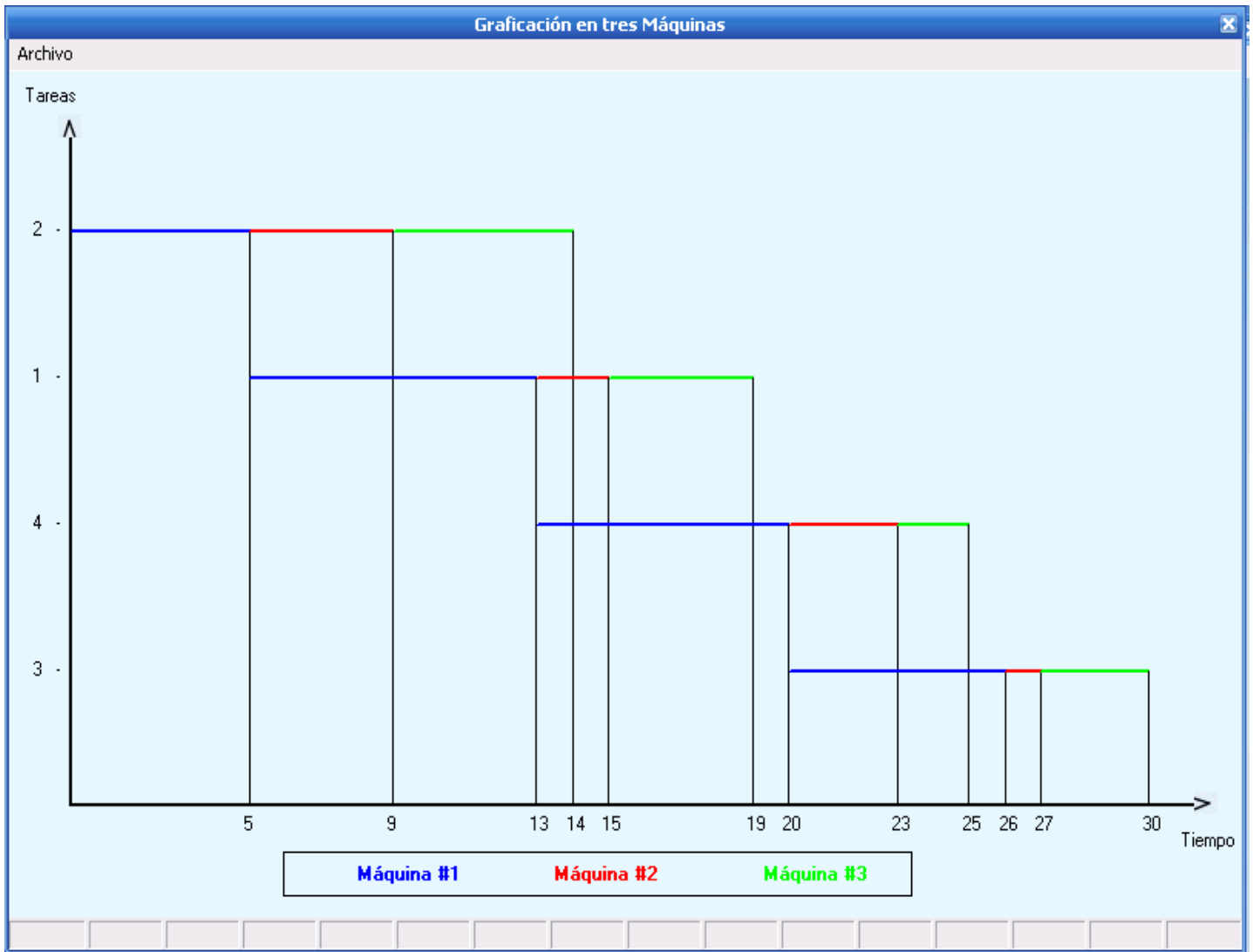


Figura 14 Ejemplo de interfaz: “Graficación de tareas en tres máquinas”

El tratamiento de errores posibilita el buen funcionamiento de una aplicación dándole una mejor apariencia ante los clientes. Dentro del tratamiento de errores tenemos errores al cargar algún fichero que no se validó en esa operación, en caso de introducir datos incorrectos, entre otros. A continuación se presenta un ejemplo.

Cuando el usuario quiere salvar los datos al fichero y estos no han sido introducidos.

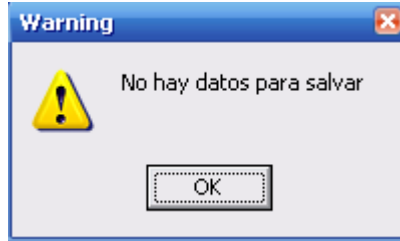


Figura 15 Ejemplo de error

### 4.6 Diagrama de Despliegue

El Diagrama de Despliegue se utiliza para modelar el hardware utilizado en las implementaciones de sistemas y las relaciones entre sus componentes.

Los elementos usados por este tipo de diagrama son nodos (representados como un prisma), componentes (representados como una caja rectangular con dos protuberancias del lado izquierdo) y asociaciones. [18]

Como el diagrama de despliegue de este software tiene un solo nodo que es una computadora personal, se considera que no se hace necesario mostrarlo en este epígrafe.

### 4.7 Diagrama de Componentes

Un diagrama de componentes representa la separación de un sistema de software en componentes físicos (por ejemplo archivos, cabeceras, módulos, paquetes, etc.) y muestra las dependencias entre estos componentes. [19]

Diagrama de Componentes

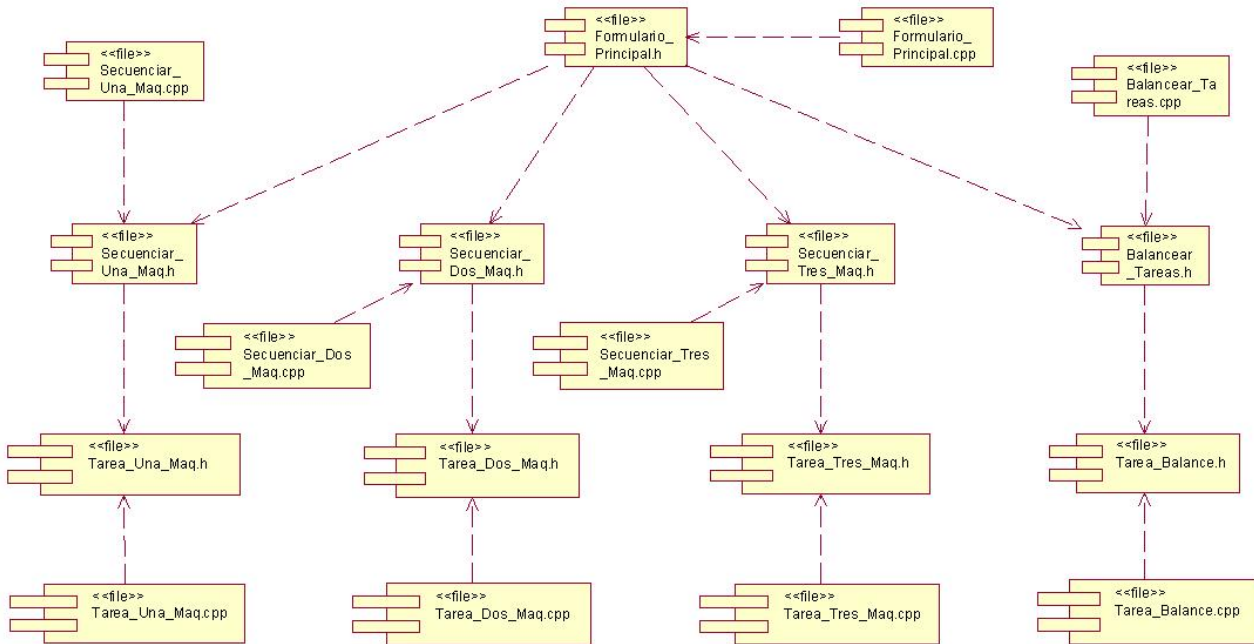


Figura 16 Diagrama de Componentes

Conclusiones

En este capítulo se finalizó la etapa de diseño e implementación del sistema, se mostraron los diagramas de clases del diseño, así como los diagramas de secuencia uno por cada caso de uso, los cuales son unos de los artefactos más importantes y que requieren de mayor dedicación en esta etapa.

Se finalizó esta etapa con el desarrollo de un diagrama de componentes, el último permitió describir los elementos físicos del sistema y la relación entre los mismos, permitiendo esto llegar al producto final.



### Conclusiones

Del estudio bibliográfico realizado para la realización de este trabajo se obtienen las siguientes conclusiones:

- Es necesario el desarrollo de una implementación informática que permita obtener la programación de la producción, específicamente para los problemas de secuenciación de tareas en máquinas y el balance de líneas de ensamblaje.
- Se realizó la versión 1.0 del software ProgProd.
- El software puede ser utilizado en empresas cuyo proceso de producción tenga las características para el cual la aplicación fue diseñado, así como soporte de ayuda en la enseñanza de estos tópicos en la carrera de Ingeniería Industrial e Ingeniería de Procesos Agroindustriales para la Tarea Álvaro Reinoso.
- El software cumple con todos los motivos planteados.

### **Recomendaciones**

Tomando como base la investigación realizada y con la experiencia obtenida durante la realización del presente trabajo, quedan algunas recomendaciones que pueden servir de punto de partida para mejorar aún más todo lo obtenido en el mismo:

- Que se continúe estudiando este tema para que se incrementen más funcionalidades a la aplicación desarrollada.
- Aplicar este software a empresas, para que se obtengan mejores resultados en su producción.
- Realizar encuestas que evalúen lo que ha representado este software para las empresas, así como para la Tarea Álvaro Reinoso (TAR).

## **Bibliografía**

- Chase, R.B., F.R. Jacobs, and N.J. Aquilano, Administración de la Producción y Operaciones para una Ventaja Competitiva. 10a. ed. 10/02/07. 848.

## Referencias Bibliográficas

1. *Tema 8 Planificación y programación de la producción.* 14/01/07.
2. Castro, W.A.S. *EL PROCESO DE PLANIFICACIÓN, PROGRAMACIÓN Y CONTROL DE LA PRODUCCIÓN. UNA APROXIMACIÓN TEÓRICA Y CONCEPTUAL.* 18/01/07 [cited; Ingeniero Industrial (CUI, 1993)  
Master en Producción (UCLV,1999)  
Candidato a Doctor en Ciencias Técnicas.(UCLV)  
Profesor Asistente de la Universidad Nacional de Colombia]. Available from:  
<http://www.gestiopolis.com/recursos/documentos/fulldocs/ger1/plaprocon.htm>
3. Alvaro, J.A.H., *La gestión de la fábrica: Modelos para mejorar la competitividad.* 2004 ed. 18/01/07.
4. *¿Qué es la planeación de la producción?* 21/01/07. p. Titulación por el Centro Universitario Villanueva, adscrito a la Universidad Complutense de Madrid y por el Instituto Superior de Educación, Administración y Desarrollo (ISEAD).
5. Mirón, S.I. *Configuraciones productivas. Conceptos y tipologías fundamentales.* 20/01/07 [cited; Ingeniero Industrial por la Universidad Central de Las Villas (UCLV).  
Diplomado en Administración y Dirección de Empresas.  
Magíster (MSc) en Creación, Estrategia y Gestión de Empresas por la Universidad Autónoma de Barcelona, España.  
Doctor (PhD) en Ciencias Técnicas por la Universidad Central de Las Villas (UCLV).]. Available from: <http://www.monografias.com/trabajos16/configuraciones-productivas/configuraciones-productivas.shtml>.
6. *SOFTWARE DE GESTION (PRODUCCION y MRP).* 24/01/07 [cited; Available from: <http://www.bowlingya.com.ar/MRP.htm>
7. *Motor de modelado.* 22/01/07 [cited; Tomado de la revista Colombia, Ciencia y Tecnología de Colciencias Vol. 23 No. 3]. Available from:  
<http://www.universia.net.co/noticias/noticiadelDia/produccionmodelo.html>
8. *TASK MANAGER.* 12/02/07 [cited; Available from:  
[http://www.softmachine.es/Soluciones/control\\_tareas.htm](http://www.softmachine.es/Soluciones/control_tareas.htm).
9. *Programación en Pascal.* 28/01/07 [cited; Available from: <http://lenguajes-de-programacion.com/programacion-en-pascal.shtml>.
10. *Programación Delphi.* 28/01/07 [cited; Available from: <http://lenguajes-de-programacion.com/programacion-delphi.shtml>.
11. *Los lenguajes de programacion mas populares.* 28/01/07 [cited; Available from:  
<http://tech.todoprogra.com/2007/04/11/los-lenguajes-de-programaci%F3n-mas-populares/>.
12. *Curso de C++.* 28/01/07 [cited; Available from: <http://usuarios.lycos.es/absurdosoyyo/tutc.html>.
13. *C++.* 28/01/07 [cited; Available from: <http://es.wikipedia.org/wiki/C%2B%2B>.

14. *La herramienta de desarrollo en C++ preferida*. 28/01/07 [cited; Available from: <http://www.abox.com/productos.asp?pid=66>].
15. *El IDE (Entorno de Desarrollo Integrado)*. 28/01/07.
16. *Proceso Unificado de Rational*. 28/01/07 [cited; Available from: [http://es.wikipedia.org/wiki/Proceso\\_Unificado\\_de\\_Rational](http://es.wikipedia.org/wiki/Proceso_Unificado_de_Rational)].
17. *Lenguaje Unificado de Modelado*. 28/01/07 [cited; Available from: <http://es.wikipedia.org/wiki/UML>].
18. *Diagrama de despliegue*. 24/04/07 [cited; Available from: [http://es.wikipedia.org/wiki/Diagrama\\_de\\_despliegue](http://es.wikipedia.org/wiki/Diagrama_de_despliegue)].
19. *Diagrama de componentes*. 24/04/07 [cited; Available from: [http://es.wikipedia.org/wiki/Diagrama\\_de\\_componentes](http://es.wikipedia.org/wiki/Diagrama_de_componentes)].

## Glosario de términos

- **Balance de líneas de ensamblaje:** Es un problema de programación de tareas en el cual se quiere determinar el número de puestos de trabajos que debe tener la línea, se desea asignar todas las tareas a estaciones de trabajo de forma que se minimice el tiempo ocioso.
- **Centro de trabajo:** Es el área de un negocio donde se realiza la organización de los recursos productivos, pueden estar formados por una sola máquina, varias máquinas, una línea de ensamblaje o un área donde se desempeñe un grupo concreto de trabajo.
- **Gráfico Gantt:** Gráfico de barras para planear y controlar trabajos.
- **Líneas de ensamblajes:** Estructura de proceso diseñada para fabricar partes separadas, las partes se mueven a una velocidad controlada, recorriendo una serie de estaciones de trabajo.
- **Peso posicional:** Es la prioridad dada a cada tarea para su ejecución en el balance de líneas de ensamblaje.
- **Relaciones de precedencias:** El orden en que deben desempeñarse las tareas en el proceso de ensamble.
- **Secuencia:** Es el proceso de determinar cual trabajo se realizará primero en una máquina o centro de trabajo. Es decir establecer el orden de prioridad en que los trabajos deben ser ejecutados.
- **Tarea (Trabajo):** Es la actividad que se realiza en un puesto de trabajo o máquina. .
- **Tiempo de ciclo:** Es el tiempo entre dos salidas consecutivas, tiempo máximo que un producto puede estar en la estación de trabajo, este tiempo tiene que ser mayor o igual al mayor tiempo de procesamientos de las tareas.

- **Tiempo de procesamiento:** Es el tiempo que se demora un producto en un puesto de trabajo, un obrero, etc.
- **Tiempo ocioso (Tiempo muerto):** Es el tiempo en que no se está utilizando el puesto de trabajo, el obrero, etc.

## Índice de Figuras

Figura 1 Diagrama de Clases del Dominio -----	35
Figura 2 Diagrama de CU del Sistema -----	38
Figura 3 Diagrama de clases del diseño -----	48
Figura 4 Diagrama CU Gestionar Tareas Una Máquina -----	49
Figura 5 Diagrama CU Gestionar Tareas Dos Máquinas -----	50
Figura 6 Diagrama CU Gestionar Tareas Tres Máquinas -----	51
Figura 7 Diagrama CU Gestionar tareas de balance -----	52
Figura 8 Diagrama de Secuencia del CU Gestionar Tareas Una Máquina -----	53
Figura 9 Diagrama de Secuencia del CU Gestionar Tareas Dos Máquinas -----	54
Figura 10 Diagrama de Secuencia del CU Gestionar Tareas Tres Máquinas -----	55
Figura 11 Diagrama de Secuencia del CU Gestionar Tareas de Balance -----	56
Figura 12 Ejemplo de interfaz: “Ventana principal” -----	61
Figura 13 Ejemplo de interfaz: “Secuenciar tareas en tres máquinas” -----	62
Figura 14 Ejemplo de interfaz: “Graficación de tareas en tres máquinas” -----	63
Figura 15 Ejemplo de error -----	64
Figura 16 Diagrama de Componentes -----	65



**Índice de Tablas**

Tabla 1	Justificación de los actores del sistema	38
Tabla 2	Descripción del CU Gestionar tareas para una máquina	40
Tabla 3	Descripción del CU Gestionar tareas para dos máquina	42
Tabla 4	Descripción del CU Gestionar tareas para tres máquina	44
Tabla 5	Descripción del CU Gestionar tareas de balance	46
Tabla 6	CE_Tareas_Balance	56
Tabla 7	CE_Tarea_Una_Maq	57
Tabla 8	CE_Tarea_Dos_Maq	57
Tabla 9	CE_Tarea_Tres_Maq	57
Tabla 10	CC_Gestor_Tareas_Balance	58
Tabla 11	CC_Gestor_Tarea_Una_Maq	58
Tabla 12	CC_Gestor_Tarea_Dos_Maq	59
Tabla 13	CC_Gestor_Tarea_Tres_Maq	59