SIMULACIÓN CON PROMODEL

Casos de producción y logística

ITM
Instituto Tecnológico Metropolitano
5044402

SIMULACIÓN CON PROMODEL

Casos de producción y logística

Luis Ernesto Blanco Rivero
Iván Darío Fajardo Piedrahíta

Segunda edición

COLECCIÓN DESARROLLO EMPRESARIAL

EDITORIAL ESCUELA COLOMBIANA DE INGENIERÍA

Simulación con ProModel. Casos de producción y logística
Primera edición: mayo de 2001
Segunda edición: marzo de 2003
Primera reimpresión: junio de 2004
Segunda reimpresión: noviembre de 2006

© Luis Ernesto Blanco Rivero e Iván Darío Fajardo Piedrahíta, 2003
© Escuela Colombiana de Ingeniería
Avenida 13 Nº 205-59
(Autopista Norte, kilómetro 13, costado occidental)
Fax: 6762655 • Bogotá
www.escuelaing.edu.co

Editorial Escuela Colombiana de Ingeniería
Telefax: 6762655 • editor@escuelaing.edu.co

Dirección editorial
Ana Mariela Ortiz Tamayo

Coordinación editorial
Jorge Cañas Sepúlveda

Diseño de portada
María Clemencia Afanador

Preprensa digital
Asercolor Ltda.

ISBN 958-8060-35-4

Impresión
Digiprint Editores

Prohibida la reproducción total o parcial de esta obra, por cualquier medio, sin autorización escrita de la Escuela Colombiana de Ingeniería.

Impreso en Colombia - Printed in Colombia

658.5
B641

A mi familia
Alba, Alba María y Luis Hernando, Luis Mauricio,
Jesús Ernesto, Gloria Lucía, Gabriel Ernesto, Julián Camilo, mis hermanos, tías y primos.

A mis inolvidables soportes espirituales
María J. Rivero vda. de Blanco, Pablo M. Castro y Marina Cáceres.

A los estudiantes de producción avanzada de la Escuela Colombiana de Ingeniería
y a todos ustedes.

Luis Ernesto Blanco Rivero

A mis padres y a Nohorita

Iván Darío Fajardo Piedrahíta

ón y logística

)06

Piedrahíta, 2003

idental)

ería
u.co

ıalquier medio, sin
ıgeniería.

Contenido

INTRODUCCIÓN XIII
AGRADECIMIENTOS XVII

1. Qué es la simulación y para qué sirve

QUÉ ES SIMULACIÓN 3
PARA QUÉ SIRVE LA SIMULACIÓN 5
PASOS PARA REALIZAR UNA SIMULACIÓN 7
Paso 1: Elaborar un plan de estudio 8
Paso 2: Definir el sistema 12
Paso 3: Construir el modelo 13
Paso 4: Ejecutar experimentos 14
Paso 5: Analizar los resultados 15
Paso 6: Reportar los resultados 15
CONCEPTOS BÁSICOS DE PROMODEL® 16
File 16
Build 21
Locations 21
Características de una Location (Sitio de trabajo) 23
Entities 27
Path Networks 28
Resources 30
Especificaciones del Recurso 32
Processing 33
Descripción 33
Operation 34
Arrivals (Llegadas) 35
Variables 36

Attributes (Atributos) 37
Simulation 38

2. Modelos para comenzar a simular con ProModel

¿POR QUÉ ESTOS MODELOS? 41
MODELO 1. BACKGROUND (EL TELÓN DE FONDO) 42
MODELO 2. ENTRADAS 45
MODELO 3. SALIDAS 50
MODELO 4. ENTRADAS Y SALIDAS 53
MODELO 5. MÁQUINAS 56
MODELO 6. VARIABILIDAD DE ENTRADAS Y MÁQUINAS 61
MODELO 7. RUTAS Y RECURSOS 66
MODELO 8. OPERARIOS Y TURNOS 70
MODELO 9. MEDICIONES DE TIEMPOS Y DE INTERVALOS EN PROCESO 76
MODELO 10. USO DE HERRAMIENTAS STAT::FIT Y GRAPHICS EDITOR 79
MODELO 11. FÁBRICA DE MONEDAS DE CENTAVO (PENNY TWO) 86
MODELO 12. MEZCLA DE MODELOS 91
MODELO 13. USO DE LAS INSTRUCCIONES LOAD Y UNLOAD 95
MODELO 14. PASTEURIZADORA DE LECHE 98

3. Casos de producción

CASO 1. TEORÍA DE RESTRICCIONES
CONCEPTOS DE *LA META* Y *FACTORY PHYSICS* 107
PARÁMETROS DE PRODUCCIÓN SEGÚN GOLDRATT 107
IMPORTANCIA DE LA SIMULACIÓN PARA SOLUCIONAR EL PROBLEMA 108
PARÁMETROS Y LEYES DE LA PRODUCCIÓN SEGÚN HOPP Y SPEARMAN 109
MODELANDO EL SISTEMA DE PRODUCCIÓN DE INDUSTRIAS ECI 111
ANÁLISIS DE LOS RESULTADOS DEL MODELO 120

CASO 2. CAJERO AUTOMÁTICO
COLAS M/M/1 123
EN QUÉ CONSISTE UNA COLA M/M/1 124
MODELANDO EL CAJERO AUTOMÁTICO CON PROMODEL® 125
ANÁLISIS DE LOS RESULTADOS 133

Caso 3. Cajeros de un banco
Colas M/M/n 137
En qué consiste una cola M/M/n 139
Modelando el banco con ProModel® 139
Análisis de los resultados 148

Caso 4. Sistemas *pull* versus sistemas *push* 161
Parámetros que se deben considerar en la comparación 164
Modelando el sistema MRP (*push*) 164
Modelando el sistema JIT o Kanban (*pull*) 173
Modelando el sistema Conwip (*pull*) 180
Modelando el sistema DBR (*pull*) 184
Análisis de los resultados 186

4. Casos de logística

Caso 1. Cargue y descargue de camiones 191
Modelando el sistema de cargue y descargue de camiones 193
Análisis de los resultados 202

Caso 2. Búsqueda y asignación de rutas 205
Modelando el sistema de asignación de rutas 206
Análisis de los resultados 209

Caso 3. Cadena de abastecimientos 211
Modelando el sistema logístico o cadena de abastecimientos 215
Presentando el sistema logístico o cadena de abastecimientos al gerente 221

Caso 4. ¿Qué es eso del crossdocking? 223
Modelando el muelle de intercambio rápido 226

Caso 5. Centro del distribución de mercancías 237
Las tres partes principales del modelo 240
Diagnóstico de la operación del centro de distribución 244
Unidades movilizadas por persona 244
Propuestas de mejoramiento 247
Conclusiones 248

Índice de figuras 249

Introducción

"Esto es un ingeniero: una persona
que hace posible nuestro confortable
vivir... quien nos guía a las estrellas
y al más allá... y para quien la realidad
tiene muy pocas fronteras.
Él es un ser muy especial".

Luis Ernesto Blanco Rivero

Este libro surge como respuesta a la necesidad sentida por estudiantes de ingeniería industrial, especialmente de una herramienta metodológica que les permita tener una mayor información, eminentemente práctica, para el manejo del ProModel®, complementaria a la ya existente en los manuales de soporte del *software*, escritos usualmente en inglés.

En la década anterior aparecieron obras y conceptos importantes en la literatura relacionada con la ingeniería industrial, específicamente en áreas de operaciones y de producción, como una visión completamente renovada de la logística, también conocida como la cadena de abastecimientos, y de la física de plantas, que utilizan enfoques de análisis diferentes de los tradicionales. Inicialmente dos situaciones reales justificaron el hecho de escribir esta obra: primero, la mayoría de la literatura mencionada se encuentra en inglés; segundo, la devaluación del peso frente al dólar causó un incremento considerable en los costos de los libros importados. Además, la acogida y los comentarios de los lectores de la primera edición llevaron a la inclusión de un nuevo capítulo más elemental.

En esta obra se pretende explicar, a través de casos, las novedades bibliográficas mencionadas y cumplir un triple propósito:

1. *Conceptual.* Presentar parámetros fundamentales de producción en la actualidad, como *Throughput, Work In Process, Cycle Time,* Ley

de *Little*, cuellos de botella, variabilidad e interdependencia, y de logística, como la cadena de abastecimientos.

2. *Metodológico*. Servir de guía al lector para que pueda entender, paso a paso, y aplicar los principales comandos del ProModel® en los casos que se presentan en el libro.
3. *Bibliográfico*. Complementar, en español, las ayudas que brindan los manuales de ProModel®.

El libro consta de cuatro capítulos: ¿Qué es y para qué sirve la simulación?, Modelos para comenzar a simular con ProModel®, Casos de producción y Casos de logística.

En el capítulo primero, ¿Qué es y para qué sirve la simulación? se mencionan y se explican cuidadosamente los pasos que deben seguirse cuando se quiere simular. Además, se muestra el panorama general del ProModel®, sus conceptos básicos, los *menús* principales, las instrucciones esenciales de un modelo y se da una explicación somera de cada una de las ventanas que aparecen a medida que se construye un modelo de simulación.

En el capítulo segundo (nuevo), Modelos para comenzar a simular con ProModel®, como su nombre lo indica, a través de catorce modelos se revisan las partes esenciales de cualquier modelo y su procedimiento tratando de habituar al lector a seguirlo, también se instruye sobre los fundamentos de la gran mayoría de los comandos de ProModel®. Adicionalmente, se explica el uso de variables y comandos que se emplean para incrementar, decrementar, esperar o moverse; así como el uso de pequeñas subrutinas. Al finalizar este capítulo, se pretende que el lector pueda elaborar modelos de mediana dificultad.

En el capítulo tercero, Casos de producción, se presentan cuatro casos típicos tomados de la realidad de los sistemas de producción: teoría de restricciones, cajero automático (colas M/M/1), cajeros de un banco (colas M/M/n) y sistemas *pull* versus sistemas *push*.

En el primer caso, mediante la simulación de una industria de manufactura, se explican los parámetros, variables y leyes a que hacen referencia los libros *La meta* y *Factory Physics*, tratando de visualizar que su aplicación mejora notablemente el desempeño de los sistemas productivos.

En el segundo caso, se explica en qué consiste una cola M/M1 y se toma un cajero automático como ejemplo de ésta, cuyo funcionamiento se simula. En el modelo se explica el uso de variables para cronometrar la hora de llegada y salida de los clientes al cajero, con el fin de calcular el

tiempo de servicio. Además, se explica el uso de gráficas, tanto ProModel® como *Excel*®.

En el tercer caso se explican las colas M/M/n, donde *n* es el número de servidores, y se simula la operación de un banco con cuatro cajeros y una recepción. En este modelo se explican los ciclos de llegada, las interfaces y el uso de recursos, entre otras instrucciones de ProModel®.

En el cuarto caso se explican las diferencias conceptuales entre los sistemas *push* representados por el MRP (*Material Requirements Planning*) y los sistemas *pull* representados por JIT (*Just In Time*), DRB (*Drum Rope Buffer*) y CONWIP (*Constant Work In Process*). Mediante simulación se muestra el mejor desempeño de los segundos comparados con los primeros, tomando como indicadores los parámetros explicados anteriormente en la física de planta: *Throughput, WIP y Cycle Time*. Además, se explica el uso de subrutinas y comandos; por ejemplo, juntarse con otra entidad.

En el cuarto capítulo, Casos de logística, se presentan cinco casos de sistemas logísticos tomados de la realidad: cargue y descargue de camiones, búsqueda y asignación de rutas, cadena de abastecimientos, ¿qué es eso del *crossdocking*? y centro de distribución de mercancías.

En el primer caso se plantean diferentes alternativas para cargue y descargue de camiones: manual, uso de bandas, uso de cargadores y combinaciones de las anteriores. Así mismo, se explican comandos de ProModel® que permiten agrupar entidades, conocer el contenido de las instalaciones y utilizar recursos durante tiempo determinado.

En el segundo caso se muestra un modelo con las principales calles y avenidas de la zona nororiental de Bogotá, se programan viajes de vehículos por una serie de sitios y se busca la mejor ruta.

En el tercer caso se ilustra el significado de logística y cadena de abastecimientos mediante un modelo que une los flujos de información, en forma de autorizaciones de transporte, y los materiales o productos terminados, como flujo de valor agregado.

En el cuarto caso (nuevo) se explica en qué consiste la técnica del *crossdocking*, utilizada por grandes cadenas de almacenes para optimizar el tiempo de entrega a los clientes y disminuir los inventarios.

En el quinto caso (nuevo) se muestra un modelo que integra algunos de los anteriores con el fin de mostrar al lector la necesidad de hacer un análisis global de los sistemas logísticos y evitar optimizaciones locales.

Agradecimientos

Los autores quieren expresar su agradecimiento y su especial reconocimiento a las siguientes personas y autoridades académicas de la Escuela Colombiana de Ingeniería, porque gracias a su esfuerzo, a su apoyo y a su fe en nosotros, fue posible escribir este libro.

A los doctores Carlos Rodado Noriega, Ignacio Pérez Vélez y Álvaro González Fletcher, en sus calidades respectivas de rector, vicerrector académico y vicerrector administrativo, en la época en que se escribió este libro, porque abrieron convocatoria de proyectos editoriales e invitaron a la comunidad universitaria a participar y, una vez aprobado el proyecto, nos asignaron el tiempo y los recursos necesarios para desarrollarlo.

Al comité editorial, a cargo de Jimena Lemoine Garzón, por su tenacidad y continua cooperación para que el proyecto se aprobara y se llevara a cabo de acuerdo con el calendario propuesto. Jimena es, sin duda, la persona a la que debemos mayor gratitud, porque nos alentó y nos ayudó efectivamente en la búsqueda de recursos en momentos críticos. Su espíritu creativo nos alienta permanentemente a continuar en esta labor.

Al equipo del Centro Editorial de la Escuela, cuyo trabajo es ejemplo de calidad, eficiencia y efectividad. A todos ellos por igual, porque trabajan coordinadamente y cada uno tiene su aporte en este libro.

A los estudiantes de Producción avanzada, de la carrera de Ingeniería industrial, que probaron los modelos de ProModel®, leyeron la edición de prueba e hicieron las observaciones pertinentes.

A los colegas del programa de ingeniería industrial, porque durante el trabajo se siente el apoyo, a veces silencioso, pero real. Y a todos aquellos que nos ayudan sin esperar ninguna mención.

LUIS ERNESTO BLANCO RIVERO
IVÁN DARÍO FAJARDO PIEDRAHÍTA

tı l, porque durante el
real. Y a todos aquellos

R STO BLANCO RIVERO
RÍ FAJARDO PIEDRAHÍTA

1

Qué es la simulación y para qué sirve

Qué es simulación

Simulación es una representación ficticia de una situación real, que se experimenta mediante modelos que son abstracciones de la realidad; el conocimiento adquirido en la simulación se aplica en el mundo real. Cuanto mayor sea el grado de aproximación de la simulación a la realidad, mayor será su utilidad. La primera acción, y requisito previo a cualquier simulación, es un buen conocimiento del sistema real. La persona que enfrenta un problema que requiere simulación para analizarlo, necesita entender muy bien las condiciones reales, sus elementos, relaciones y metas, e imaginarlas como un sistema.

Se entiende por sistema un conjunto de elementos que interactúan con el fin de lograr un objetivo común. De esta manera, un sistema de producción o de operaciones es un conjunto de personas, máquinas, materiales, conocimientos, capital y servicios que interactúan para lograr un objetivo básico; por ejemplo, *la producción de un bien o un servicio*. Hay que imaginar las fábricas, a la manera de Alex Rogo[1], como inmensas cajas negras a las que entran materiales y de las que salen productos terminados. Es necesario tener una imagen mental y el mapa del proceso productivo con sus operaciones. Es imprescindible conocer muy bien la tecnología, entendida ésta como máquinas, conocimientos, herramientas, y tener muy claro el papel que desempeñan las personas y la forma como se relacionan y trabajan para cumplir con *el objetivo común* de producir bienes que se puedan vender. Se necesita, además, conocer los parámetros que regulan el sistema: *throughput,* inventarios y gastos de operación[2].

1 Eliyahu Goldratt y Jeff Cox, *La meta, un proceso de mejora continua*, 7a. ed., Monterrey, México, Ediciones Castillo, 1998, p. 45.
2 *Ibid.*, p. 74.

Como la complejidad de los sistemas reales es generalmente muy grande, es necesario construir un modelo simplificado del sistema real, que tenga en cuenta los parámetros, variables y relaciones que sean de interés para la persona que quiere simular la realidad, con el fin de experimentar –en dicho modelo– las alternativas probables de solución al problema. Es importante que el modelo refleje muy bien el sistema real, porque los conocimientos que se obtengan de la experimentación se aplicarán posteriormente en él.

La simulación puede verse como un campo o laboratorio experimental de ciertas condiciones o relaciones del sistema real, cuyo ensayo sería muy costoso o imposible de llevar a cabo en la realidad. Si el modelo no se comporta como se espera, con determinadas condiciones experimentales, habrá oportunidad de cambiarlas, incluso varias veces, aprendiendo cada vez más sobre el modelo y, más adelante, sobre el sistema real.

Esta herramienta de análisis ha venido ganando importancia con el desarrollo de los computadores. Hasta hace muy pocos años, la simulación era tediosa por la cantidad de datos numéricos que arrojaba y por la gran cantidad de tiempo de computador que consumía. Con el avance exponencial de los microprocesadores en lo relacionado con su capacidad de procesamiento, unido al gran incremento en la capacidad de almacenamiento y a la facilidad de manejo de imágenes y de gráficas, los programas actuales de simulación animada no se parecen en nada a los ya mencionados. Los programas actuales son accesibles, fáciles de entender y traen bastante ayuda gráfica, tanto que a veces se olvida que el análisis de las condiciones, variables, relaciones y parámetros que se quieren estudiar son más importantes que el modelo animado.

La animación, por una parte, y la inmensa capacidad de procesamiento de los programas de simulación actuales, por otra, permiten el análisis de sistemas muy complejos que se pueden ir construyendo paso a paso, agregando y mezclando uno a uno sus subsistemas. El método de construcción de modelos va de lo simple a lo complejo. Esto evita, en gran parte, una de las frecuentes causas de abandono de la simulación como herramienta de trabajo: la pérdida del analista dentro de la complejidad de los datos.

Como se comentó, la simulación y el modelaje de sistemas son herramientas que ayudan a entender el comportamiento de los sistemas reales. A veces algunos usuarios se quedan únicamente en el perfeccionamiento del modelo o en el embellecimiento de su presentación gráfica, olvidándose del objetivo fundamental: el análisis de las condiciones del sistema real.

Se pueden simular variables continuas, como el flujo de crudo a través de una refinería, o variables discretas, como la llegada de automóviles a una estación de gasolina; también hay simuladores físicos o modelos a escala para simular condiciones, como las plantas piloto o las represas a escala. En el contexto de este libro únicamente nos ocuparemos de modelos con variables discretas.

En conclusión, la simulación es una herramienta de análisis de sistemas complejos, que bien utilizada puede generar desde ahorros considerables de dinero hasta el mejoramiento de la planeación y control de los sistemas productivos, pasando por el descubrimiento de muchas de las restricciones reales del sistema.

Para qué sirve la simulación

Hacer ensayos en los sistemas productivos o logísticos reales es prácticamente imposible por los altos costos y los riesgos que genera. Un ejemplo que puede sustentar esta afirmación es fruto de la experiencia profesional de uno de los autores: en una empresa dedicada a la recolección, el transporte y la entrega de mercancías, se tenía la sospecha de que con una mejor programación de los camiones, los operarios y los centros de distribución, se podía lograr un ahorro considerable de dinero. Debido al alto costo de ensayar cambios de programación, al riesgo de demora en la entrega de las mercancías a los clientes y a la resistencia de las personas al cambio, las pruebas no se podían hacer en el sistema real. Se hizo un modelo del sistema, que incluía las rutas nacionales de transporte, la recolección y distribución local, al igual que el cargue y descargue de los diferentes tipos de vehículos. Al simular el sistema, se confirmó que se podía realizar una mejor programación de los camiones, lo cual permitía un uso más racional de ellos y generaba ahorros considerables en los costos. Adicionalmente, se descubrieron los cuellos de botella que se formaban debido a la mala programación. Convencidos los directivos de la empresa de la autenticidad de los datos que arrojaba la simulación, contrastándolos con la realidad, se tomó la decisión de efectuar los cambios y de ensayar la programación en el sistema real, confirmándose los resultados obtenidos en la simulación.

La función de planeación, programación y control de la producción es una de las tareas complicadas de la empresa porque requiere mucho tiempo y utiliza modelos complejos de la investigación de operaciones o costosos

sistemas computacionales como el MRP (planeación de materiales), el MRP II (planeación de recursos), el DRP (planeación de la distribución), sin conseguir los resultados esperados. La razón principal de su falta de precisión en la programación es el hecho de ignorar que hay dos fenómenos subyacentes en los sistemas de producción y operaciones: la variabilidad y la dependencia. Como lo explican Goldratt, en *La Meta,* y Hopp y Spearman en *Factory Physics*[3], los métodos tradicionales de programación de producción y de operaciones fallan porque, en su mayoría, no consideran las *fluctuaciones estadísticas* a las que están sometidas prácticamente todas las operaciones de un proceso y porque no tienen en cuenta que éstas son interdependientes. El efecto de las variaciones estadísticas es acumulativo y afecta con mayor intensidad las últimas operaciones del proceso, las más *dependientes* de las demás, mientras que la máxima variabilidad en las operaciones se refleja, de manera acumulativa, en la variabilidad total del sistema. En la programación de operaciones, se deben combinar las herramientas, partiendo de la más indicada para el análisis: la simulación.

Como los métodos tradicionales de la investigación de operaciones no permiten observar parámetros fundamentales en la producción manufacturera, como los cuellos de botella y el *WIP (work in proccess* o inventario en proceso), se deben potenciar con los resultados de la simulación, si se quiere aumentar su utilidad.

Por otra parte, la importancia de sistemas y software comercial, como *MRP, MRPII, DRP*, no está tanto en el control y en la programación, sino en la información, porque dichos paquetes requieren datos de entrada de los procesos, de sus tiempos estándar, de las secuencias de las operaciones, de los costos, etc., de manera que su utilidad se incrementa con los conocimientos adicionales que proporciona la simulación de los sistemas productivos o de operaciones.

Otra de las ventajas de la simulación es que obliga al usuario a formarse una idea de la "totalidad" del sistema y a buscar las interrelaciones de sus subsistemas, al contrario de la mayoría de los métodos matemáticos, que optimizan localmente. Esta es precisamente la idea de la *física de planta*[4], que consiste en buscar las leyes que gobiernan el comportamiento dinámico de las plantas industriales. Los parámetros clave y básicos de la

3 Wallace J. Hopp y Mark L. Spearman, *Factory Physics, Foundations of Manufacturing Management*, Boston, Irwin-McGraw-Hill, Inc., 1996, p. 6.

4 *Ibid.*, p. 221.

dinámica de las plantas, como tasa del cuello de botella, tiempo global de procesamiento de la línea, WIP crítico y coeficiente de congestión[5], se pueden observar muy bien en simulaciones de sistemas productivos o de operaciones.

Entre las facilidades que ofrecen programas de simulación como el ProModel®, está la de poder trabajar, como se había comentado, desde lo simple hacia lo complejo, esto es, comenzando por la simulación de susbsistemas pequeños que se van uniendo y mezclando, paso a paso, hasta obtener el sistema total, de manera que se puedan tener las visiones locales y la visión total del sistema.

Pasos para realizar una simulación

Para realizar una simulación hay que identificar primero el producto o servicio que se va a simular. Un proceso de simulación se puede entender como un proyecto compuesto por tareas y recursos requeridos.

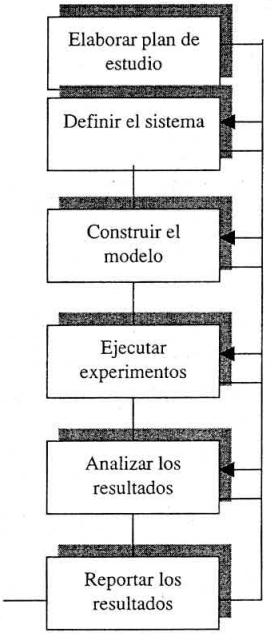


Figura 1.1 Pasos para realizar una simulación[6].

5 *Ibid.*, p. 225.
6 ProModel Corporativo, *User's Guide*, 1996, p. 36.

El éxito en la simulación dependerá de la planeación y de comprender los requerimientos necesarios. Un modelo de simulación requiere un minucioso análisis de los resultados obtenidos, buena comunicación y poseer habilidades de ingeniería para visualizar las interrelaciones del sistema. La decisión de entrar a simular un proceso se toma cuando necesitamos diseñar un nuevo sistema o modificar el existente.

No hay una regla que diga cómo simular, pero los siguientes pasos nos dan una idea al respecto.

Paso 1: Elaborar un plan de estudio

Muchos proyectos de simulación fallan debido a la mala definición de sus objetivos, lo cual crea falsas expectativas del proyecto o no deja entender en forma correcta los requerimientos. La planeación de la simulación involucra las siguientes tareas:

- Definir los objetivos
- Identificar las limitaciones o restricciones
- Conocer las especificaciones
- Desarrollar la planeación y definir resultados

a) Definir objetivos

Es necesario tener claros los objetivos que se persiguen. El objetivo de la simulación debe ser claro y es importante saber hasta dónde se quiere llegar con la simulación.

Existen diferentes tipos de modelos de simulación. Algunos se utilizan solo una vez y de desechan, mientras que otros se van enriqueciendo a medida que se van utilizando y que se van produciendo cambios dentro del proceso. Los objetivos de estos modelos son diferentes. Éstas son algunas preguntas que se deben responder para poder formular los objetivos:

- ¿Qué y cómo desea medir el desempeño de la simulación?
- ¿Quién utilizará el modelo?
- ¿A quién se le presentará la información obtenida del modelo?
- ¿Qué información se espera obtener del modelo?
- ¿Qué tan importante es la decisión que se espera tomar a partir del modelo?

b) Identificar limitaciones o restricciones

Los objetivos son tan importantes de definir como las limitaciones que puede tener el estudio que se pretende realizar.

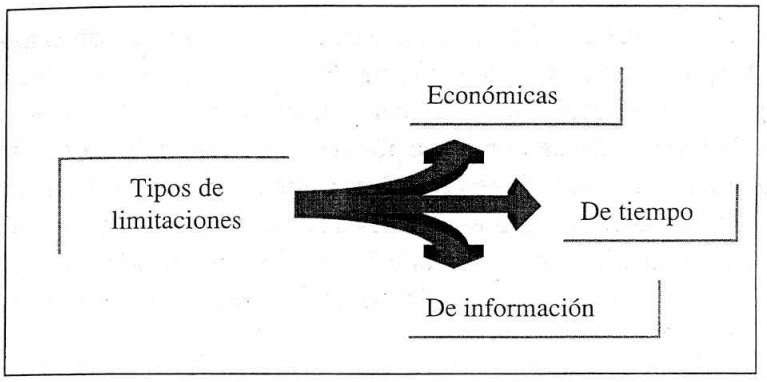


Figura 1.2 Principales limitaciones en un estudio.

Las principales limitaciones están relacionadas con los recursos económicos de que se dispongan durante el proyecto, el tiempo con que se cuenta para realizar la simulación y, una de las más importantes, si se posee toda la información del modelo que se pretende simular.

Cualquier recurso, relación, etc., que pueda tener efectos limitantes debe considerarse como restricción del sistema. Las siguientes son algunas de las preguntas que deben plantearse para identificar limitaciones del estudio.

- ¿Cuál es el presupuesto para el estudio?
- ¿Cuáles son las habilidades que se poseen para realizar el estudio?
- ¿Se tiene acceso a la información requerida?
- ¿Qué tipo de computador se utilizará en el estudio?

c) Conocer las especificaciones

Al tener claros los objetivos y las limitaciones, se deben conocer los requerimientos o especificaciones de la simulación. La definición de las especificaciones es esencial para el proyecto, desde el punto de vista de tiempo y costo.

- Alcance
- Nivel de detalle
- Grado de exactitud

Alcance

El alcance se refiere a delimitar hasta dónde se quiere llegar con el modelo, ya que debe tener un comienzo y un final. Por ejemplo, si se está simulando un centro de distribución de mercancía y se desea simular "la operación"; ésta puede abarcar desde la recolección de la mercancía hasta la distribución de la misma, o simplemente el descargue de la mercancía dentro del centro de distribución. Son dos alcances distintos, con dos inicios y finales diferentes. Es muy importante saber con anticipación qué se pretende obtener en el modelo. El alcance va directamente relacionado con el nivel de detalle necesario.

Nivel de detalle

El nivel de detalle es un factor determinante para la simulación. Las especificaciones de la duración o de la capacidad del computador en el cual se está trabajando son factores determinantes en la simulación. Cuanto más detallado sea el modelo, más espacio ocupará en el disco duro y en la memoria. Determinar el nivel de detalle es vital, ya que la velocidad de ejecución del modelo también se verá afectada. De la misma forma, un nivel de detalle insuficiente ocasionará que el modelo no sea realista, debido a la exclusión de algunas de sus variables críticas.

Más adelante se ilustra cómo afecta el nivel de detalle el tiempo de desarrollo (figura 1.3).

Grado de exactitud

Para algunos modelos o actividades, no se requiere que la información sea exacta ya que el grado de exactitud está determinado por el objetivo del estudio. La exactitud está directamente relacionada con la confiabilidad de la información.

El grado de exactitud de un modelo puede tener un impacto significativo en el tiempo y esfuerzo requeridos para obtener los datos, y en la realización del modelo. De él dependerá, en gran parte, la precisión de los resultados.

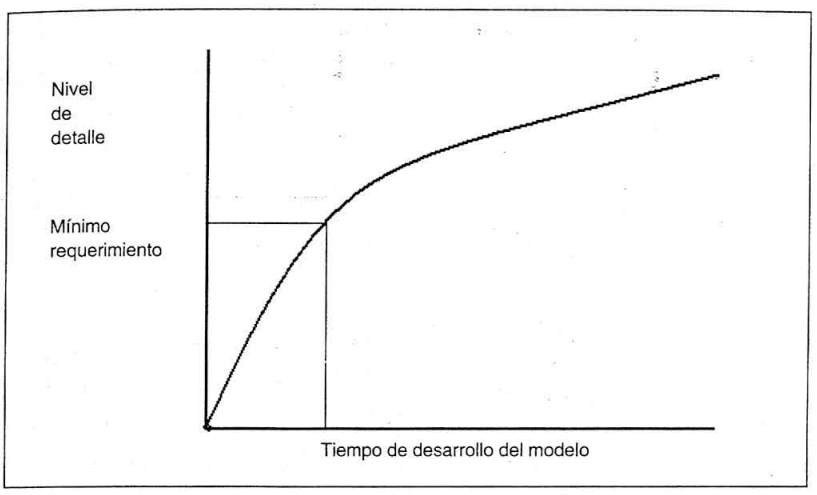


Figura 1.3 Grado de afectación del tiempo de desarrollo causado por el nivel de detalle[7].

d) Desarrollar planeación y definir resultados

La forma de presentación de los resultados es trascendental, debido a que éstos son la razón principal de la simulación. Obtener resultados que ayuden a tomar decisiones, y que sean confiables, puede involucrar gran cantidad de esfuerzo y tiempo.

Con las limitaciones y objetivos claros se debe empezar a trabajar en lo que se quiere medir o mejorar. Obviamente, el tiempo en el cual un modelo se esté simulando por completo variará, dependiendo del tamaño y de la dificultad del proyecto. Si la información no está disponible en su totalidad, probablemente el proyecto se podrá demorar más tiempo de lo que se había presupuestado al principio.

La planeación para realizar una simulación debe basarse en una proyección realista del tiempo requerido, teniendo en cuenta los siguientes aspectos:

- La definición del sistema para realizar el modelo puede tomar más de 50% del tiempo total del proyecto.
- La construcción del modelo usualmente tomará el menor tiempo dentro del proyecto, alrededor de 10 a 20% del tiempo total del mismo.

7 ProModel Corporation, *User's Guide*, 1996, p. 42.

Paso 2: Definir el sistema

Después de definir con claridad los objetivos y la organización del proyecto, se debe entrar a definir detenidamente el sistema, para lo cual hay que tener presentes los siguientes aspectos:

- Determinar la información requerida.
- Usar apropiadamente las fuentes de información.

a) Determinar la información requerida

El primer paso para definir el sistema es tener claro el proceso que se desea simular. Después de comprender el proceso, se debe determinar *en dónde* se va a recolectar la información y *quién* la suministrará. La siguiente es información importante que posiblemente necesitará:

1. ¿Cómo está distribuida la planta (*layout*) y cuáles son sus dimensiones?
2. ¿Cuántos centros de trabajo (*locations*) tiene el proceso?
3. ¿Cuáles son los tiempos de proceso de cada centro de trabajo?
4. ¿Cuántos productos o subproductos (*entities*) se quieren incluir en el modelo?
5. ¿Quién requiere un recurso y cuándo y dónde lo necesita?
6. ¿Qué se desea medir?
7. ¿Cuáles serán los indicadores que se deberán tener en cuenta?
8. ¿Cuánto tiempo se simulará el proceso?
9. ¿Qué tan variable es el proceso? ¿Se podría estandarizar?
10. ¿Quién es la persona que utilizará el programa de simulación?

Estas son sólo algunas de las preguntas que usualmente se deben formular para empezar a simular. Es necesario recolectar toda la información del proceso, por lo que se debe contar con el apoyo de la gerencia para realizar este trabajo. Como en todo sistema, si se alimenta el modelo con basura, éste arrojará basura. Esta etapa de recolección de información es muy importante, ya que la veracidad del modelo estará determinada por la veracidad de dicha información.

b) Usar apropiadamente las fuentes de información

Es importante saber de dónde se tomará la información y si la información que ya existe está actualizada para el modelo. La información se puede obtener de fuentes como las siguientes:

- Los diagramas de procesos
- Los estudios de tiempos
- Los planos de la planta
- Los diagramas de flujo
- Los pronósticos de mercadeo
- La historia que se tenga en reportes o en otro tipo de formato.

En algunos casos será necesario realizar suposiciones acerca de algunos datos, pero se aconseja utilizar el menor número de ellas. Cuanto menos suposiciones tenga el modelo, estará más cercano a la realidad. Es importante también, en lo posible, estandarizar los tiempos.

Paso 3: Construir el modelo

Una vez se tenga la información, para empezar a construir el modelo, se deberán dimensionar el proceso y el alcance que tendrá la simulación. El modelo se deberá ajustar progresivamente, por lo que se le podrán hacer mejoras en el proceso, en los tiempos, en las actualizaciones de los datos, etc. Por esta razón es fundamental definir muy bien el alcance que se quiere lograr con el modelo.

Cuando se está construyendo un modelo grande, es muy útil dividirlo en etapas. Estas etapas deben definirse claramente, para no tener problemas en el momento de analizar los resultados obtenidos.

Cuando se decide elaborar un modelo por etapas, éstas se pueden realizar en diferentes épocas o simultáneamente y por distintas personas. Es necesario que cada etapa tenga unas condiciones iniciales y unas finales. Esto es útil para trabajar en forma paralela, y evitar problemas cuando se decida ensamblar el modelo. Entre las condiciones que se deben definir para las etapas están: nombre de variables, entidades, puestos de trabajo (*locations*) y atributos.

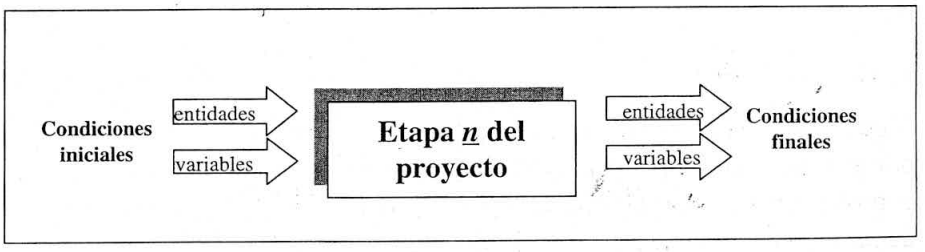


Figura 1.4 Proyecto por etapas.

Esta información será de gran ayuda en el futuro, cuando se desee realizar un cambio al modelo, u otra persona desee consultarlo.

Conceptos Básicos de ProModel®

Al abrir el programa de ProModel® el usuario se encontrará con una pantalla como la siguiente:

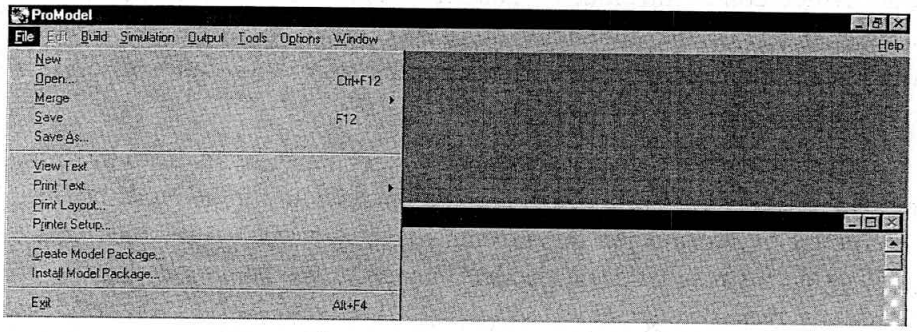


Figura 1.5 Menú Archivos.

File

New

Si existe algún modelo abierto en el momento, ProModel® pregunta si desea salvarlo y lo cierra para abrir uno nuevo.

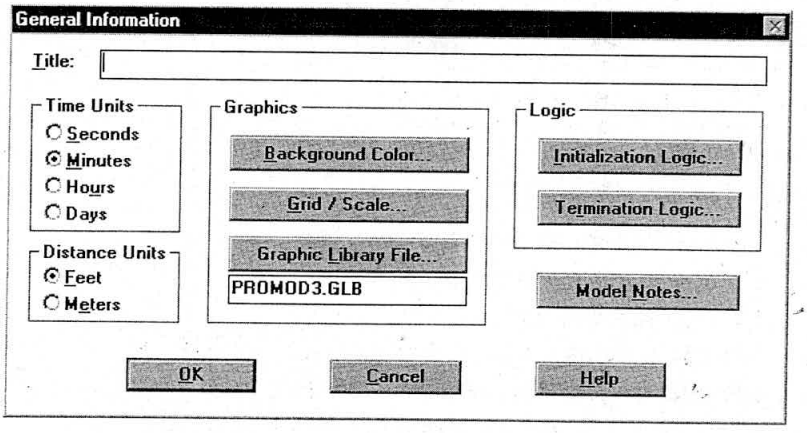
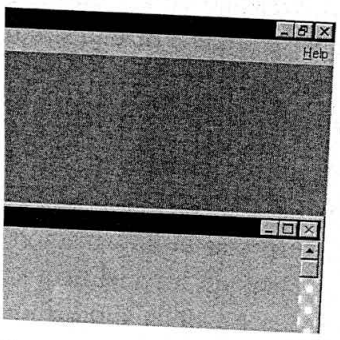


Figura 1.6 Menú Información General.

l futuro, cuando se desee rea-
see consultarlo.

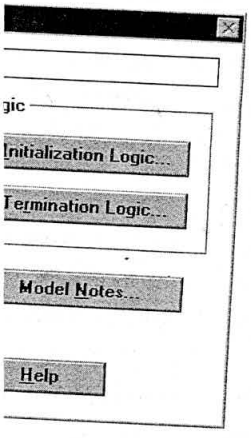
Model®

) se encontrará con una pan-



s.

ProModel® pregunta si



ral.

Cuando se crea un nuevo modelo es necesario ponerle un título y especificar en unidades el *tiempo* y la *distancia* con que se desea trabajar. Se pueden definir algunas especificaciones como:

- ***Background Color...***
  Colores de contorno que se utilizarán.
- ***Grid /Scale...***
  Se puede especificar a qué escala se va a trabajar (esta opción se estudiará con más detalle mas adelante).
- ***Graphics Library Files...***
  El nombre del archivo de la librería que se desea utilizar.
- ***Initializacion Logic...***
  Instrucciones que se imparten antes que el modelo inicie su ejecución.
- ***Termination Logic...***
  Instrucciones que se dan después que el modelo termine.
- ***Model Notes...***
  Se pueden colocar notas al comienzo del modelo. Por ejemplo: nombre de quien hizo el modelo, o una breve explicación del mismo.

Open

Abre un modelo ya creado. Si existe un modelo cargado en memoria lo cierra.

Import

Importa un modelo a ProModel PC construido en una versión 5.0 o anterior.

Merge

Mediante esta opción se pueden fusionar dos o más modelos en uno solo. Existen dos tipos de Merge.

- Model
- Submodel

- **Model**

Se pueden unir dos o más modelos independientes en uno solo.

Así mismo, une las entidades y atributos que tengan el mismo nombre y los vuelve uno solo. Si cada modelo emplea una librería diferente, une las dos librerías en una sola, con un nuevo nombre.

- **Submodel**

Esta opción se utiliza cuando un modelo está compuesto por diferentes partes o submodelos (por ejemplo, una estación de trabajo).

Cuando se fusiona un submodelo, ProModel® le coloca un identificador con todas sus características. Por ejemplo: *entities*, *locations, variables, arrays*, etc. El identificador lo puede colocar el usuario antes (*prefix*) o después (*suffix)* del nombre de las características.

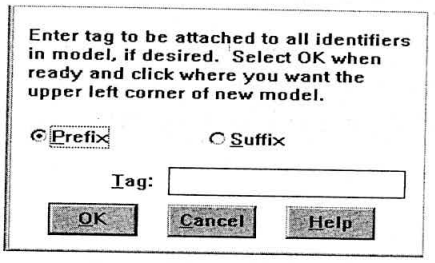


Figura 1.7 Menú Fusión o Mezcla.

En donde dice *tag* el usuario puede colocar el identificador del modelo que se va a fusionar.

Save

Sirve para salvar, en una unidad específica, el modelo cargado actualmente en memoria, con el mismo nombre con que se le guardó inicialmente.

Saves As

Se usa para salvar, en una unidad específica, el modelo cargado actualmente en memoria. Además, da la opción de cambiar el nombre.

View Text

Presentación preliminar, en pantalla, de toda la información del modelo, como *locations, arrivals, processing*.

Print Text

Muestra la información que aparece en la presentación preliminar de un archivo o en la impresora. Si se desea guardar esta información en un archivo se ejecuta la opción ***To Text File,*** y se creará un archivo con extensión .TXT. Si se desea imprimir, se elige la opción ***To Printer***.

Print Layout

De igual forma se puede imprimir el *layouí* del modelo actual. Da la posibilidad de imprimir los gráficos de contorno, *locations*, *path networks*, *resources* y *variables*.

ProModel® ofrece una herramienta para comprimir y descomprimir los modelos

Create Model Package

Con esta opción se puede comprimir el modelo que se encuentra cargado.

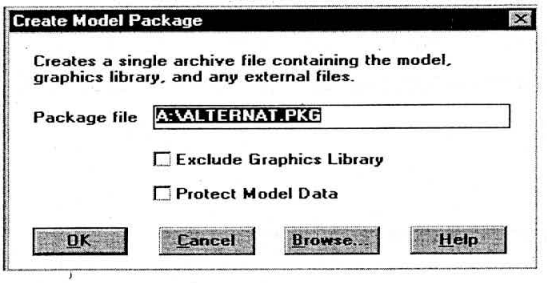


Figura 1.8 Menú para empaquetar.

Sólo se necesita indicar en dónde se desea guardar la compresión. Si se quiere, no se graban las librerías para que el tamaño del archivo disminuya. Mediante la opción *Protect Model Data,* se puede proteger el código del programa para que no pueda ser alterado. El archivo quedará grabado con una extensión .PKG

Install Model Package

Con esta opción se puede descomprimir un modelo.

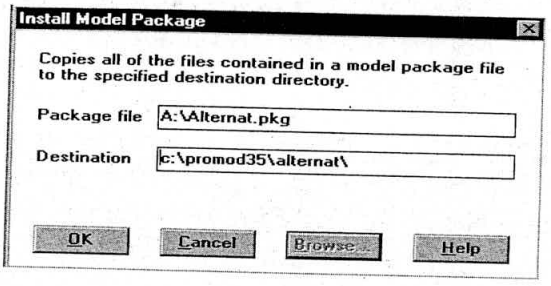


Figura 1.9 Menú para instalar modelo empaquetado.

Para instalar un archivo .PKG sólo se indica la ruta donde se encuentra el archivo y donde se desea grabarlo.

Edit

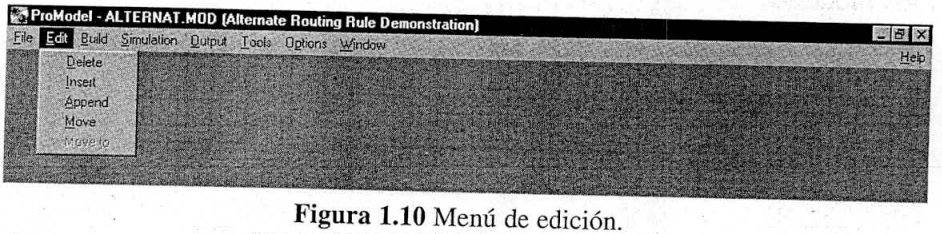


Figura 1.10 Menú de edición.

El menú está compuesto básicamente de cinco instrucciones.

Delete

Elimina el registro seleccionado de una tabla.

Insert

Inserta un registro en una tabla donde se le indique.

Append

Agrega un registro al final de la tabla.

Move

Mueve el registro marcado a la posición que se le indique con la instrucción ***Move To.***

Build

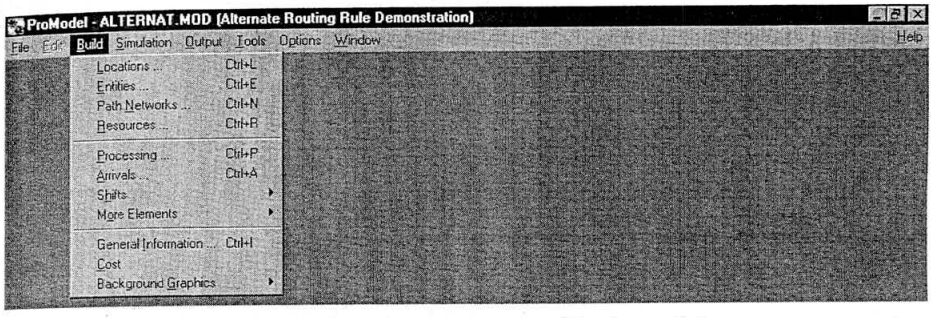


Figura 1.11 Menú para construcción de modelos.

El menú *Build* es uno de los más utilizados, ya que en él se especifican las características del modelo; por ejemplo: *locations, entities, resources.*

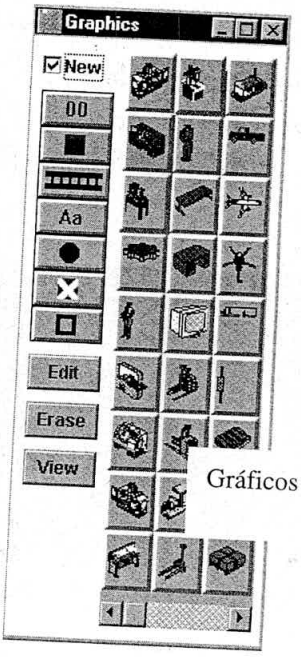
Locations

Las *locations* representan el lugar donde la entidad realizará un proceso o algún otro tipo de actividad. Por ejemplo, en un modelo de una empresa manufacturera la *location* es un torno, o una prensa, o una fresadora; para crear *locations* se cuenta con una librería de gráficos.

En primer lugar, se selecciona con el *mouse* la que desea, luego se selecciona en el *layout* y se hace clic con el *mouse.*

Cuando se crea una *location*, automáticamente se crea un registro con las características que se muestran en la figura 1.13. Éstas son: *Icon, Name, Cap., Units, DTs..., Stats..., Rules..., Notes...*

Nuevo registro _____
Contador _____
Indicador de capacidad _____
Colas o bandas _____
Texto _____
Indicadores de estado _____
Entidad / Ubicación _____
Región _____
Editar un gráfico _____



Gráficos de la librería

Figura 1.12 Menú de gráficos.

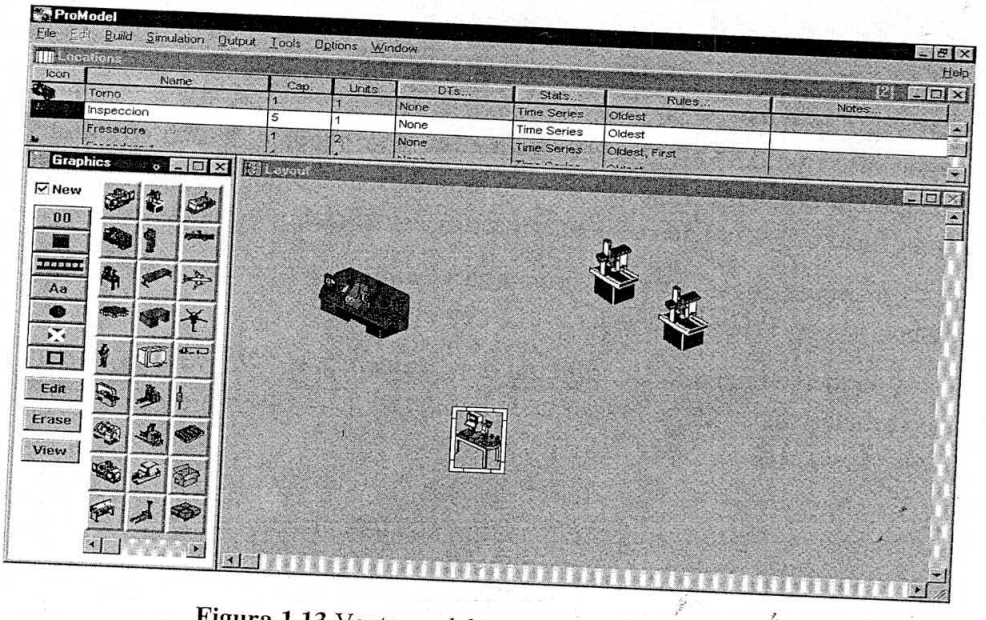
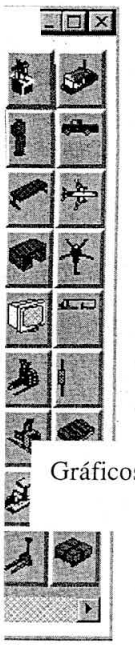
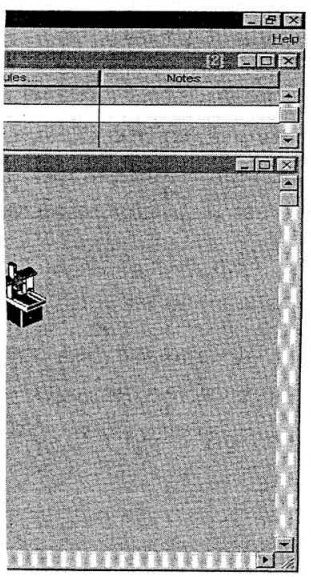


Figura 1.13 Ventanas del menú de sitios de trabajo.



Gráficos de la librería



le trabajo.

Características de una Location (Sitio de trabajo)

Icon

Aparece el gráfico de la *location* que se seleccionó.

Name

Es el nombre que se puede dar a la *location*.

Cap.

La capacidad de la *location* se refiere al número de entidades que puede procesar al mismo tiempo. Por ejemplo, un punto de inspección puede revisar cinco piezas simultáneamente, pero generalmente una máquina sólo puede procesar una sola pieza a la vez.

Unit

Una *location* puede repetirse varias veces dentro de un proceso. Por ejemplo, en un banco existen tres cajeros que tienen el mismo trabajo con las mismas características.

DTs

Define los tiempos muertos que puede tener una *location*.

Figura 1.14 Menú de tiempos de parada.

Un tiempo muerto es una parada de la operación que debe hacer una *location*. Existen cuatro tipos de paradas:

a) Clock

Esta parada se ejecuta después que la entidad ha cumplido cierto tiempo específico. Por ejemplo, en un banco un cajero empieza a trabajar a las ocho de la mañana y tendrá un descanso a las diez de la mañana durante 15 minutos.

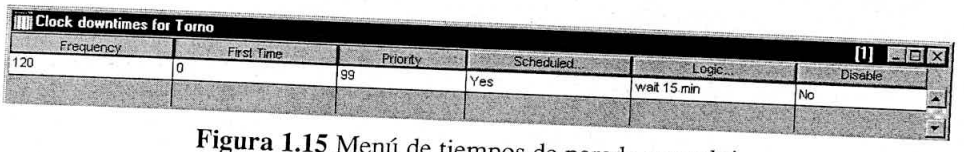


Figura 1.15 Menú de tiempos de parada por reloj.

Frequency

Tiempo en el cual está presupuestada la parada.

First Time

Tiempo desde el cual empieza a contabilizarse el tiempo que se le ha dado en *frequency*.

Priority

Prioridad que el tiempo muerto tiene para que suceda.

Schedule...

Si la parada está contemplada dentro de la programación del proceso y no desea que se tenga en cuenta en las estadísticas, se coloca ***Yes***; de lo contrario, la parada se considerará en los resultados de la *location*.

Logic

Se pueden crear instrucciones o procesos que ocurren en el momento de la parada.

Disable

Seleccionar ***Yes*** si se desea que la parada no se tenga en cuenta en la simulación, y **No** si se desea que se efectúe la parada.

b) Entry Downtime

Esta parada se realiza después de un número determinado de entradas a la *location*. Continuando con el ejemplo del banco, el cajero puede parar un momento después de haber atendido a 30 personas.

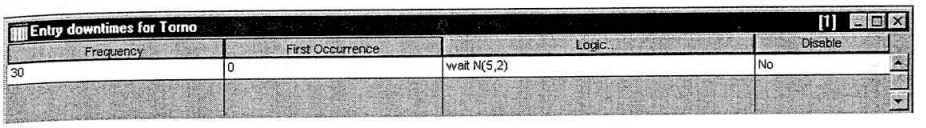


Figura 1.16 Menú de tiempo de parada por entradas.

Frequency

Número de entidades que deben ser procesadas antes de realizar la parada.

First Ocurrence

Tiempo desde el cual empieza a contabilizarse el número de entradas que se le ha dado en *frequency*.

Logic

Permite crear instrucciones o procesos que ocurren en el momento de la parada.

Disable

Seleccionar ***Yes*** si se desea que la parada no se tenga en cuenta en la simulación, y ***No*** si se desea que se efectúe la parada.

c) Usage Downtime

Esta parada es muy parecida a la del *Clock*, pero difiere en que el tiempo empleado es tiempo efectivo o de uso de trabajo. Por ejemplo, si el cajero entró a las ocho de la mañana, puede descansar sólo después de haber trabajado una hora efectiva. Esto quiere decir que sólo va a descansar cuando complete una hora efectiva de trabajo atendiendo a los clientes, no importa si acaba a las diez o a las nueve de la mañana.

Paso 4: Ejecutar experimentos

Cuando ya se tiene el modelo, es importante ejecutarlo varias veces, cambiándole las condiciones para cerciorase que el modelo representa la realidad y no únicamente una situación puntual. Todo modelo debe contemplar la variabilidad del proceso y ésta se debe estudiar con precisión. El resultado de ejecutar una sola iteración de la simulación implica que ésta solo representará una posible situación entre las muchas que pueden existir. Se requiere realizar múltiples *replicaciones* en el modelo, verificando sus resultados.

Para ejecutar varios experimentos, es necesario definir el tiempo que durará la simulación, el número de *replicaciones* por ejecutar, y tener claro el objetivo y la precisión de los resultados de la simulación.

Existen dos tipos de simulaciones:

- Terminales
- No terminales

a) Terminales

Modelos en los cuales el proceso está definido en el tiempo. El comienzo y el final están perfectamente determinados por un estado o por un número. Por ejemplo, a una ceramista le encargan 30 figuras. Como el modelo tiene un final determinado, se detendrá cuando haya realizado las 30 figuras. Para elaborar un modelo terminal es necesario responder algunas preguntas:

1. ¿Cómo debería ser el estado inicial del modelo?
2. ¿Cuál es el evento que ocasionará la finalización del modelo?
3. ¿Cuántas *replicaciones* deberán hacerse?

El número de *replicaciones* se podrá determinar por la precisión que se desee en los resultados.

b) No terminales

Modelos en los cuales el proceso nunca termina. Para finalizar el proceso, por lo general debe determinarse el tiempo de la simulación. Teóricamente el proceso nunca termina, aunque puede tener interrupciones como cambio de turnos, tiempos muertos, etc. Un ejemplo es una empresa de manufactura que produce clavos. Este es un proceso que teóricamente nunca termina,

ya que siempre la empresa está fabricando su producto. Cuando se cuenta con un modelo no terminal es necesario tener claro el tiempo de calentamiento (*warmup*) y el tiempo que durará la simulación.

Paso 5: Analizar los resultados

Después de tener el modelo y de ejecutarlo, se pasa a analizar los resultados alcanzados. Esta fase es una de las más importantes, ya que se debe ser muy crítico con la información que arroja el modelo. El mayor beneficio que se obtiene de una simulación no es necesariamente encontrar verdades absolutas, sino lograr que el experimento muestre cómo funciona el sistema simulado.

Infortunadamente, los errores se detectan cuando se analizan los resultados y no cuando se está ejecutando la simulación. Es fundamental saber interpretar la información que arrojan los programas, ya que mediante el análisis de los resultados se pueden identificar los cuellos de botella del proceso y de la simulación.

Paso 6: Reportar los resultados

Para finalizar, es necesario realizar un reporte con los resultados. La presentación de resultados es, muchas veces, tan importante como la información que contienen los reportes; por eso es necesario hacer un informe conciso, que utilice el mayor número de gráficos, ya que éstos explican más que un simple número. Los resultados deben presentarse en términos fáciles de comprender y de evaluar, y en ellos han de proponerse recomendaciones acerca del sistema, basadas en los resultados de la simulación. Crear una base de datos de los resultados para tener una historia de la evolución del sistema que se ha simulado es de gran utilidad.

En el modelo es muy importante documentar aspectos como los siguientes:

- De qué manera se construyó
- De dónde se sacó la información
- Cómo almacenar la información original
- Documentar las rutinas del programa (para qué sirve, etc.)
- Identificar variables y atributos utilizados
- Identificar a las personas que realizaron la simulación.
- Establecer la fecha de la simulación.

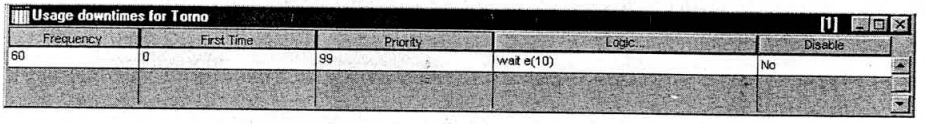


Figura 1.17 Menú de tiempo de parada por uso.

Frequency

Tiempo efectivo en el cual está presupuestada la parada.

First Time

Momento desde el cual empieza a contabilizarse el tiempo que se le ha dado en *frequency*.

Priority

Prioridad que el tiempo muerto tiene para que suceda.

Schedule...

Si la parada está contemplada dentro de la programación del proceso y no desea que se tenga en cuenta en las estadísticas coloque ***Yes***; de lo contrario, la parada se tendrá en cuenta en los resultados de la *location*.

Logic

Permite crear instrucciones o procesos que ocurren en el momento de la parada.

Disable

Seleccionar ***Yes*** si se desea que la parada no se tenga en cuenta en la simulación, y ***No*** si desea que se efectúe la parada.

d) Setup Downtime

Puede utilizarse en situaciones en que la *location* procesa diferentes tipos de entidades, pero necesita un alistamiento en cada corrida.

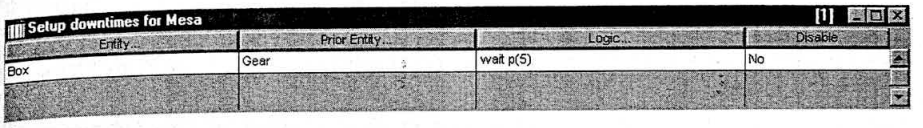


Figura 1.18 Menú de tiempos de alistamiento.

Entity

Entidad que se está procesando en el momento en la *location*.

Prior Entity...

Nueva entidad que se va a procesar en la misma *location*.

Logic

Permite crear instrucciones o procesos que ocurren en el momento de la parada (por ejemplo, el tiempo de aprestamiento o actividades que se hacen entre una entidad y la otra).

Disable

Seleccionar ***Yes*** si se desea que la parada no se tenga en cuenta en la simulación, y ***No*** si se quiere que se efectúe la parada.

- **Stats**

Al hacer clic en esta opción, aparecen tres niveles:

- ***None***. No se recogen estadísticas.
- ***Basic.*** Solamente se muestra la utilización y el promedio del tiempo.
- ***Time Series***. Calcula y grafica información del modelo en relación con el tiempo.

Entities

La entidad es la materia prima de un producto o servicio que va ser procesada en una *location*. Las entidades se pueden transformar durante la simulación. Por ejemplo, en un centro de distribución, la entidad principal es

una caja, pero las cajas se pueden agrupar (*Group*) en una estiba, que es otra entidad. Cada entidad tiene sus propiedades.

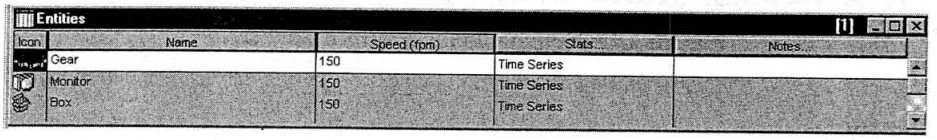


Figura 1.19 Menú para entidades.

- **Icon**

Aparece el gráfico de la entidad seleccionada.

- **Name**

Nombre que se puede dar a una entidad.

- **Speed (mpm/fpm)**

Velocidad a la cual la entidad se desplaza dentro del modelo. El valor predeterminado es 150 fpm o 50 mpm (este dato es opcional).

- **Stats**

Al hacer clic en esta opción, aparecerán tres niveles:

- ***None***. No se recogen estadísticas
- ***Basic.*** Solamente se muestra la utilización y el promedio del tiempo.
- ***Time Series***. Calcula y grafica información del modelo con respecto al tiempo.

Path Networks

Todo proceso necesita un diagrama de recorrido para los recursos (personas, máquinas, etc.) que se utilizan para movilizar las entidades (papeles, piñones, cajas, entre otros).

También se usa para informar al modelo cuánto tiempo se demora de una *location* a otra. ProModel® tiene la capacidad de calcular de manera automática la distancia de una *location* a otra.

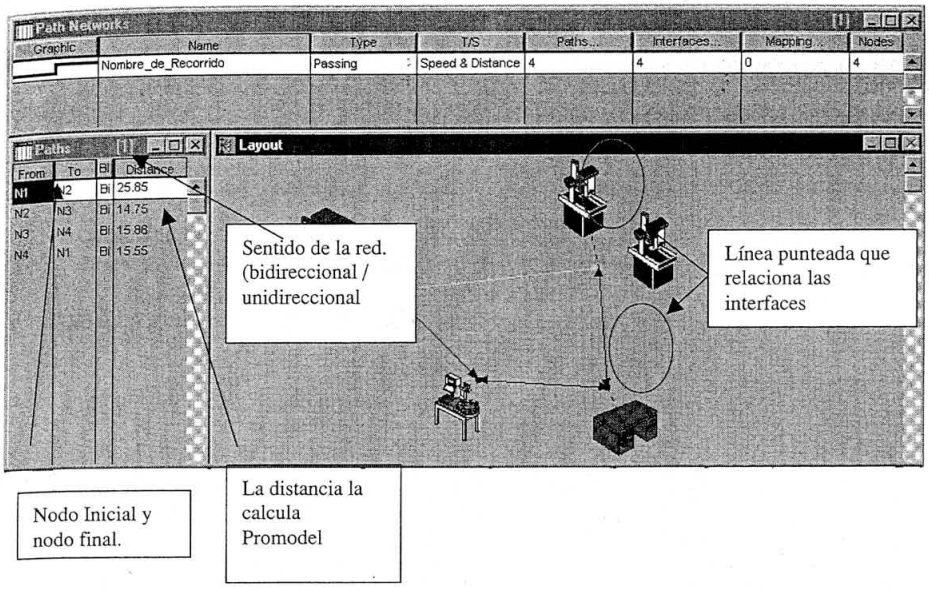


Figura 1.20 Ventanas de las redes de recorridos.

Graphic

Puede cambiar el color a la red o dejarla visible o invisible durante la simulación.

Name

Nombre que se da a la red que se crea.

Type

Hay tres tipos de redes: *Passing, Non-passing y Crane*. La *Passing* se utiliza cuando los recursos se pueden mover libremente por la red, sin ningún tipo de restricción. *Non-passing* es una red que tiene restricciones; por ejemplo, un tipo de vehículo que no puede adelantar. *Crane* es especial para grúas y puentes grúas.

T/S

Se puede trabajar con dos tipos de unidades básicas de distancias: en tiempo o velocidad y distancia.

Interfaces

Cada *location* debe tener un nodo que la relaciona. La forma de unir los nodos de la red con las *location* es mediante las interfaces. Al hacer clic en *Interfaces* aparece el siguiente cuadro:

Figura 1.21 Menú Interfaces.

Ejemplo: Al hacer clic en el nodo N1 y luego en la *location* Torno, debe crearse una línea punteada, como se muestra en la figura 1.20.

Mapping

Cuando se tenga una red conformada por varios nodos, ProModel® automáticamente buscará el camino más corto para ir del nodo N1 al nodo N2. Si no se está de acuerdo con el camino propuesto por ProModel®, se podrá proponer uno diferente para desplazarse del nodo N1 al N2.

Resources

Un recurso puede ser una persona, un equipo o un vehículo que puede desempeñar o realizar diferentes operaciones a las entidades, como transporte de un nodo a otro y operaciones puntuales a la entidad.

Para utilizar los recursos, previamente se debió definir una red (*Path Network*).

Units

Número de recursos que desea tener. Esta cantidad es fija, es decir, no puede variar durante la simulación.

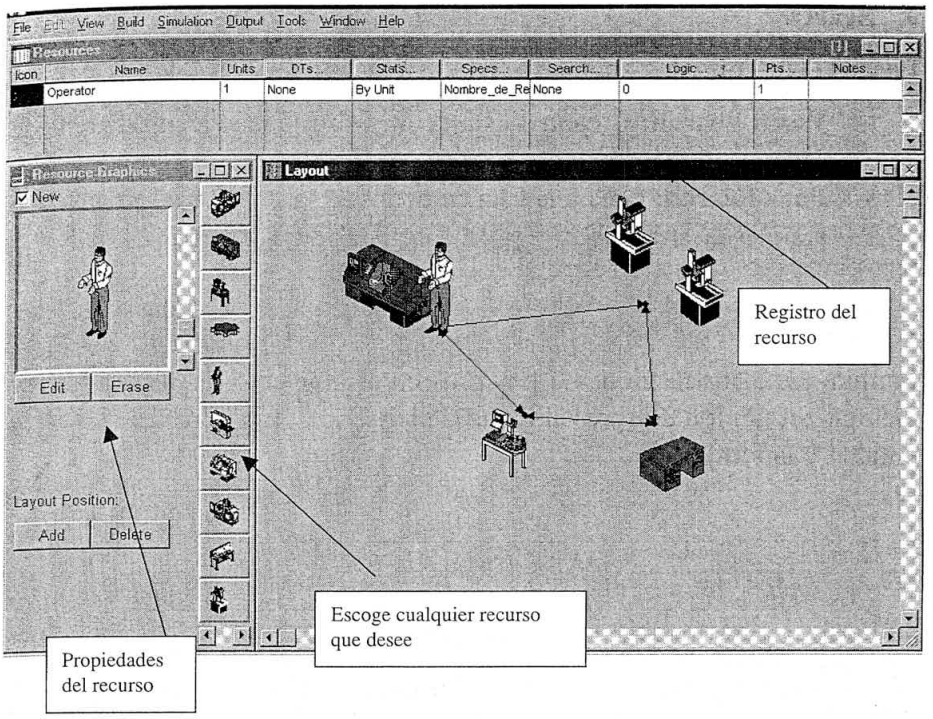


Figura 1.22 Ventana para colocar recursos.

- **DTs...**

El recurso puede tener paradas al igual que las *locations*.

- **Stats**

Al hacer clic en esta opción, aparecen tres niveles:

- ***None.*** No se recogen estadísticas.
- ***Basic.*** Solamente se muestra la utilización y el promedio del tiempo.
- ***Time Series.*** Calcula y grafica información del modelo en relación con el tiempo

Specs...

En este campo se pueden asignar al recurso una red y otras propiedades (figura 1.23).

- **Search**

Puede elegir entre dos opciones: *Work Search* y *Park Search.*

- **Work Search:** Crea una lista de *locations* donde la entidad puede esperar un recurso.
- **Park Search:** Crea una lista de nodos a los cuales pueden enviarse si no están trabajando y están esperando otro trabajo.

- **Logic**

Permite crear instrucciones o procesos en el momento de la parada (por ejemplo, el tiempo de aprestamiento o actividades que se hacen entre una entidad y la otra).

Especificaciones del Recurso

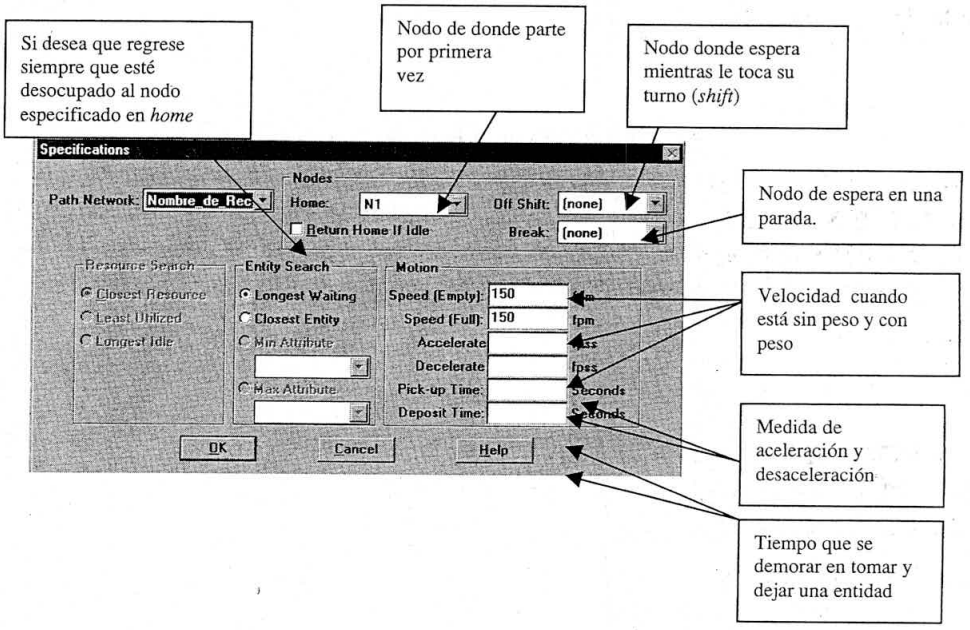


Figura 1.23 Ventana para especificaciones del recurso.

En el cuadro de las especificaciones, se puede introducir al recurso diferentes tipos de información; por ejemplo:

Processing

El menú de proceso es uno de los más importantes debido a que en él se programa la operación. Normalmente todo proceso tiene un diagrama de proceso o de operaciones; esta información se transcribe del diagrama al computador.

Antes de diseñar el proceso se deben crear entidades, recursos, *locations* y *path network* que necesita el modelo. Se debe de tener claro cómo es el proceso que se va a simular.

Descripción

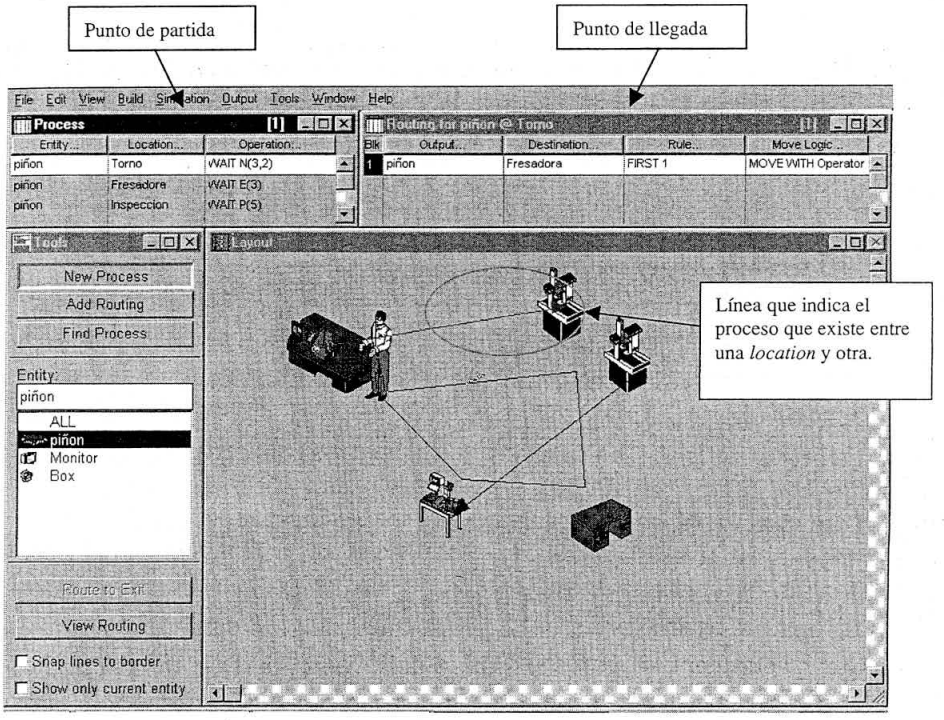
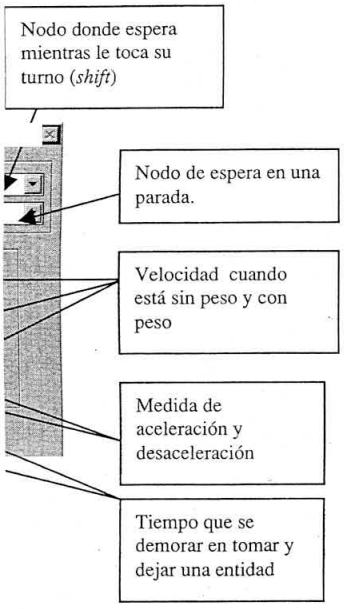


Figura 1.24 Menú para definir el proceso.



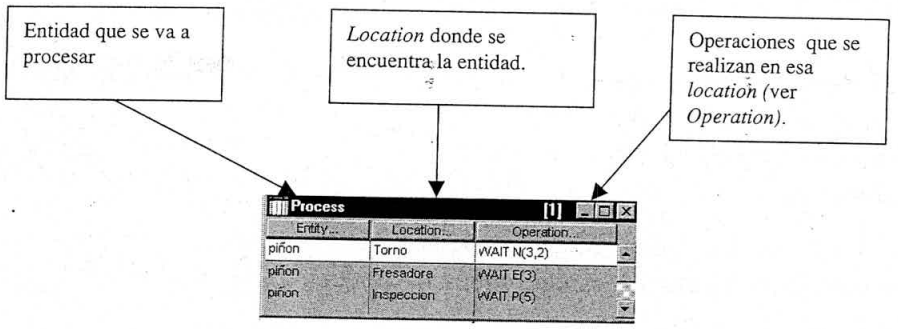


Figura 1.25 Ventana para definir la entrada al proceso.

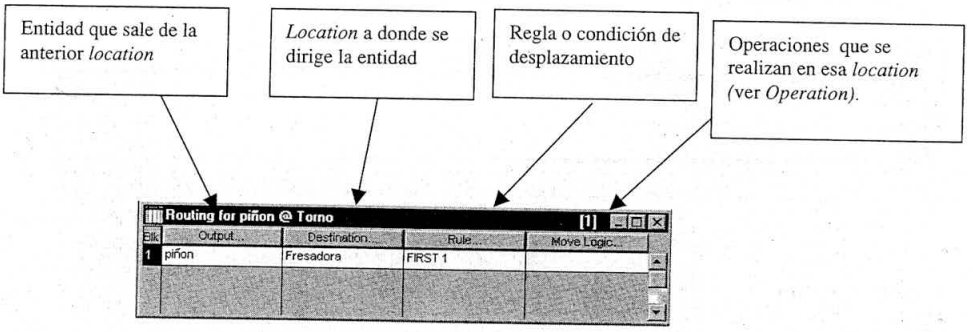


Figura 1.26 Ventana para definir la salida del proceso.

Operation

Dentro del cuadro de *Operation* se puede crear un gran número de instrucciones para simular los procesos. Por ejemplo, agrupar determinado número de entidades, realizar operaciones con variables, colocar tiempos de procesos, mandar mensaje por pantalla.

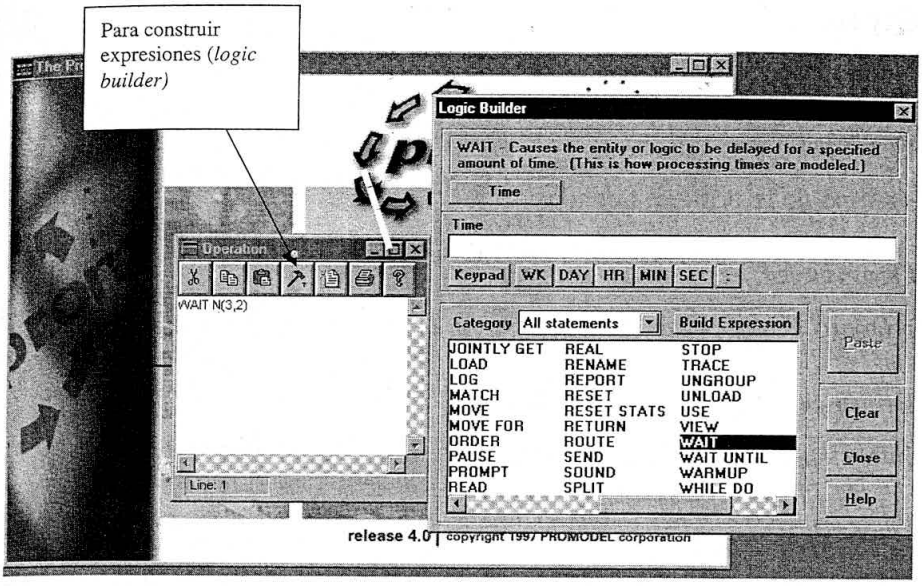


Figura1.27 Ventana para colocar instrucciones.

Arrivals (Llegadas)

Todo sistema tiene un punto de partida a donde llega material (información, personas, etc.) para que el proceso pueda empezar a desarrollarse. Si tomamos como ejemplo un banco, éste necesita que los clientes *(entities)* lleguen a determinada hora, en cierta cantidad y con determinada frecuencia al proceso para que puedan fluir a través de él.

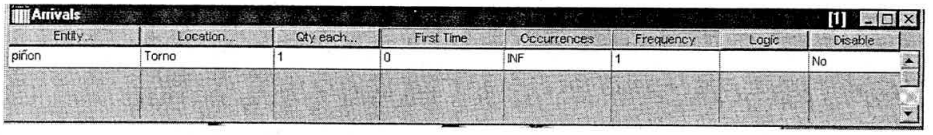


Figura 1.28 Ventana para definir llegadas.

Cuando una entidad nueva llega al sistema, se produce un *arrival*.

Entity

Nombre de la entidad que llega al sistema.

Location

Nombre de la *location* a donde llega la entidad.

Qty each...

Número de entidades que llegan a la vez (lote).

Fist Time

Tiempo de la primera llegada.

Ocurrences

Número de veces que puede llegar una entidad de esas características.

Frequency

Intervalo de tiempo entre una llegada y otra.

Logic

En este campo se pueden definir operaciones que se ejecuten en el momento de la llegada de la entidad al sistema.

Disable

En este campo se activa o desactiva la llegada al sistema.

Variables

Se pueden emplear variables de tipo global y local. Las variables son útiles para calcular o guardar información numérica, ya sea real o entera. El valor de una variable global se puede utilizar en cualquier parte de la simulación, mientras que el de una variable local sólo se podrá utilizar dentro del bloque (*logic*) en el que se colocó.

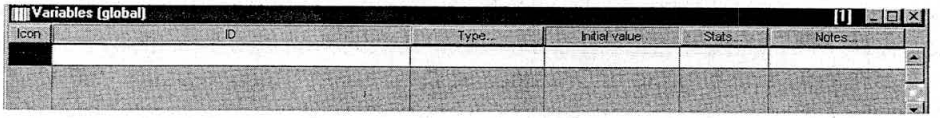


Figura 1.29 Ventana para definir variables.

Icon

Si el campo dice ***Yes***, muestra las variables por pantalla; de lo contrario no lo hará.

ID

Nombre que se da a la variable.

Type

La variable puede ser de tipo entera o real.

Stats

Al hacer clic en esta opción, aparecerán tres niveles:

- ***None***. No se recogen estadísticas.
- ***Basic.*** Solamente se muestra la utilización y el promedio de tiempo.
- ***Time Series***. Calcula y grafica información del modelo con relación al tiempo.

Attributes (Atributos)

Los *atributos* son muy parecidos a las variables. El atributo es un tipo de variable del sistema que solo guarda la información en memoria de una *location* o entidad particular. Los atributos pueden contener números reales y enteros. Con los atributos se pueden diferenciar entidades. Por ejemplo, si se está simulando el descargue de dos tipos de vehículos, una tractomula de 30 toneladas y un furgón de 12 toneladas, el procedimiento para descargar un vehículo u otro es el mismo, sólo varía la cantidad de mercancía que puede transportar cada vehículo. El atributo se puede utilizar para almacenar la cantidad de mercancía que trae cada camión (entidad). La diferencia entre un atributo y una variable es que el atributo guarda información de cada entidad que llega al sistema, mientras que la variable no. El atributo es una condición inicial, una marca, que se puede dar a una entidad o a una *location* y después no se puede modificar, mientras que las variables si pueden cambiar su valor dentro del proceso.

Simulation

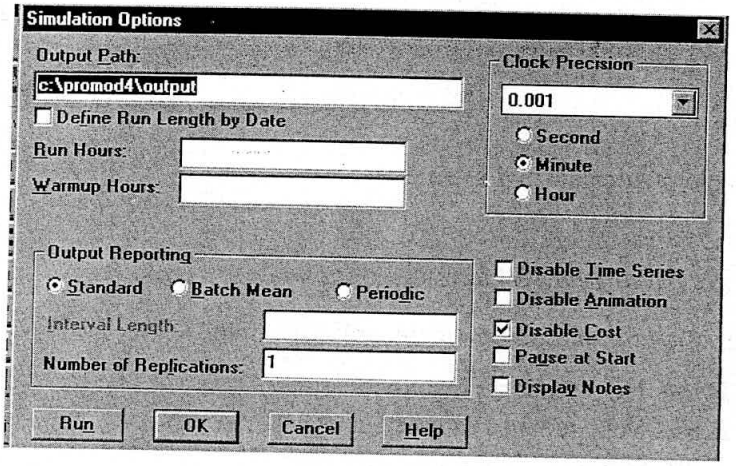


Figura 1.30 Menú para definir condición de la simulación.

En este menú se pueden ejecutar los modelos. Para ello hay que entrar en *Options* y configurar el modelo.

Output Path

Ingresa el lugar donde se desea almacenar el modelo. Utilizar en lo posible, C:\Temp.

Define Run Length by date

Si desea que el modelo corra en determinada fecha calendario, hay que elegir esta opción. Es necesaria cuando se asignan turnos *(shifts)*.

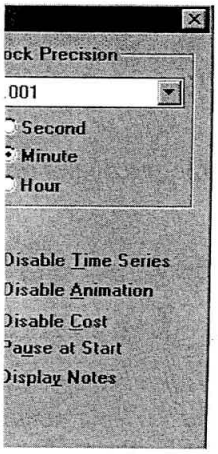
Run Hours

Tiempo en horas que se va a simular.

Warmup Hours

Tiempo en horas de precalentamiento.

Hay otras opciones que se explicarán a medida que se utilicen; por ejemplo, cómo desactivar la animación si no se necesita. El modelo corre mucho más rápido sin animación. También se puede activar el análisis de costos cuando se requiera.



la simulación.

Para ello hay que entrar

lelo. Utilizar en lo posi-

ha calendario, hay que
turnos *(shifts)*.

da que se utilicen; por
cesita. El modelo corre
le activar el análisis de

2

Modelos para comenzar a simular con ProModel

¿Por qué éstos modelos?

La construcción de flujos de materiales a través de un proceso de producción es una de las actividades importantes de la simulación con ProModel®. El objetivo que perseguimos los autores al incluir estos modelos es brindar al lector la oportunidad de modelar flujos en procesos de producción, desde lo más simple hasta lo medianamente complejo. Se inicia con la ambientación gráfica de un sistema productivo, que sirve como telón de fondo para los modelos siguientes. A continuación se revisa la entrada de materiales en un sistema productivo, la forma de diferenciarlos y de contarlos. Después se muestran formas alternativas de salida de un sistema productivo y la manera de incluir en la simulación el efecto gráfico de un proceso de transformación.

Una de las grandes ventajas del ProModel® es poder incluir tiempos de parada en las máquinas, que suceden por diferentes causas. Esta ventaja se ilustra en uno de los modelos. La variabilidad y la dependencia son dos fenómenos que están siempre presentes en los sistemas productivos; por tanto se quiere que el lector aprenda a incluir la variabilidad en las entradas, los tiempos de operación, las paradas de las máquinas por mantenimiento, las pausas de descanso de los operarios.

La medición de los tiempos de operación, de espera y de ciclo es otro de los aspectos que se incluye en los modelos. De igual manera se miden los inventarios en proceso, la tasa de salida de productos en cada operación y en el proceso total. Paso a paso, a través de la modelación, se van presentando aspectos teóricos avanzados de producción.

También se ilustra el uso de herramientas de ProModel®, como el *Stat::Fit*, que sirve para encontrar la función de distribución que más se

ajusta a una serie de datos reales, y el *Editor Gráfico*, que sirve para mejorar ostensiblemente la ambientación gráfica de los modelos.

Modelo 1. Background (El telón de fondo)

Con este modelo se pretende aprender a hacer la ambientación de una empresa, que contiene una bodega con cuatro oficinas separadas por divisiones; en cada una de ellas, situadas contra la pared del fondo, hay un oficinista. También existen cuatro tornos que van al frente, al lado izquierdo, sobre una isla que los separa de los otros centros de trabajo. Cada uno de los tornos tiene un operario a su lado. Al frente, al lado derecho, hay tres fresas sobre una isla que las separa de los centros de trabajo. Al lado de cada máquina hay un operario. En el muro de fondo, la bodega tiene dos ventanas y, encima de éstas, el nombre de la empresa, que es INDUSTRIAS ECI. En la pared izquierda de la bodega hay una puerta que da al parqueadero donde hay una isla, y en ésta hay cuatro vehículos, estibas, tubos, cajas de cartón y dos montacargas, uno en la puerta y otro en el parqueadero.

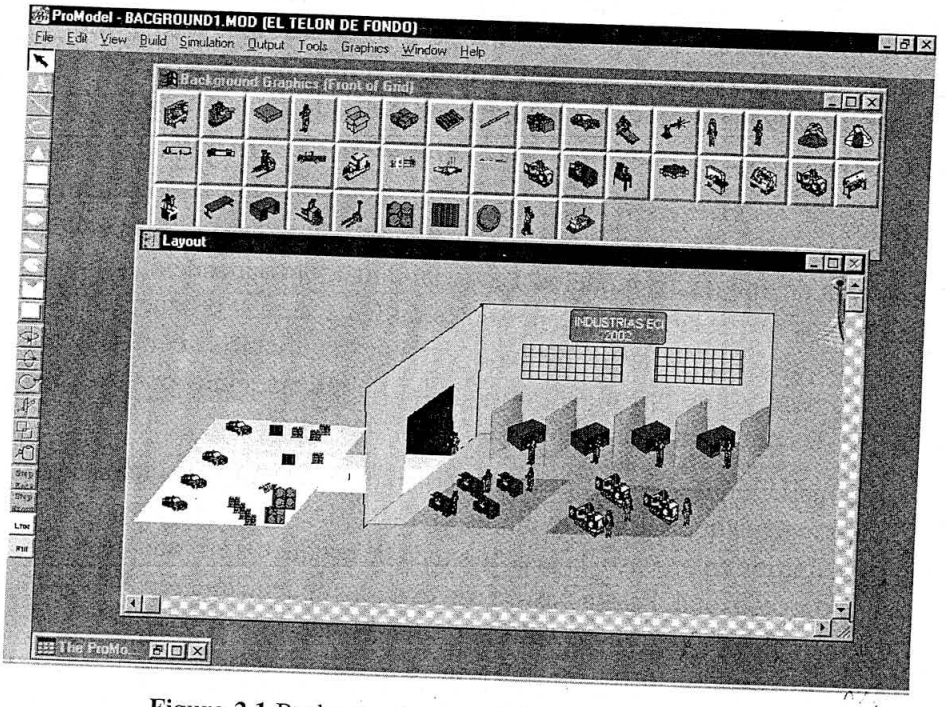
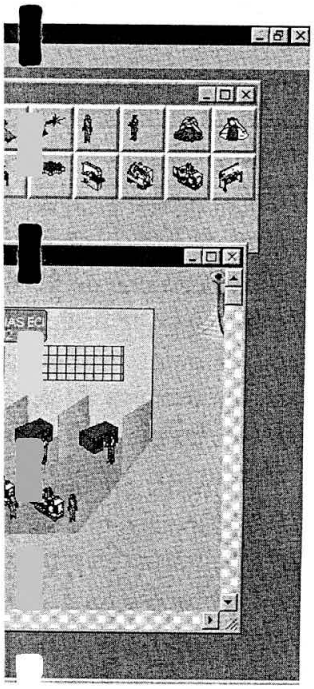


Figura 2.1 Background de una industria manufacturera.

if ɔ, que sirve para mejo-
os modelos.

le ıondo)

a bientación de una em-
ıa separadas por divisio-
d del fondo, hay un ofici-
f nte, al lado izquierdo,
s ... trabajo. Cada uno de
al lado derecho, hay tres
o de trabajo. Al lado de
ɔnıo, la bodega tiene dos
mpresa, que es INDUS-
h / una puerta que da al
cuatro vehículos, estibas,
en la puerta y otro en el



anufacturera.

Para entrar a ProModel® comience en *Inicio* y busque *Programas-ProModel* 4.22 (o ProModel 2002). Cuando lo encuentre, coloque el indicador encima, siga la dirección de la flecha y haga clic sobre el nuevo letrero de *ProModel 4.22* (o ProModel 2002). Si está trabajando con versión estudiantil, aparece un letrero que advierte sobre la capacidad. Pulsar *OK* para cerrar la ventana que aparece a continuación, haciendo clic sobre la X de la esquina superior derecha. Aparecen entonces dos ventanas: una grande llamada *ProModel*, y una interior llamada *Layout*. Éste siempre será el comienzo para cualquier modelo.

Iniciar haciendo clic en *File-New-Title* y escribir El Telón de Fondo en el renglón correspondiente al título. Verificar que en *Graphic Library File* aparezca PROMOD4.GLB, definir *Time Units* en Minutes y *Distance Units* en Meters.

Para salvar, hacer clic en *File-Save As* y, con la flecha que apunta hacia arriba, en la parte derecha del renglón que dice *Save in* (Models), buscar el ícono del disco (C:)\ *Temp*, guardar en el archivo "Background-su nombre" y hacer clic en *Save*. Esta operación debe repetirla con *File-Save* cada cinco minutos para asegurarse de ir guardando su diseño y no esperar el final para salvar, puesto que puede tener alguna sorpresa desagradable.

Ahora, ir a *Build-Background Graphics- Front of Grid*. Aparece una nueva ventana (*Background Graphics (Front of Grid)*) con máquinas y una barra de herramientas de dibujo.

Imagine ahora la ventana *Layout* dividida en cuatro partes resultantes al trazar imaginariamente una recta vertical por la mitad y otra recta horizontal por la otra mitad.

Seleccione el *rectángulo* de la barra de dibujo y colóquelo en el cuadrante imaginario superior derecho, cuyas dimensiones aproximadas son 10x5 cm. Para cambiar el color, haga clic en la parte inferior de la barra de dibujo, que aparece coloreada con el mismo color del rectángulo, y que dice *fill*. Aparece una paleta de colores. Seleccione un azul claro. El rectángulo debe ser, ahora, azul claro. Esta será la pared posterior de la planta. Para mejorar la precisión, utilizar *View-Zoom*; para retornar al original, *View-Zoom To Fit Layout*.

Para construir el piso de la planta, seleccionar el *trapecio* en las herramientas de dibujo (parece una M). Hacer clic con el botón izquierdo sobre la esquina inferior izquierda del rectángulo que ya construyó; luego, hacer clic de nuevo con el botón izquierdo en la esquina inferior derecha del

rectángulo. Desplazarse 5 cm hacia abajo y 2 cm a la izquierda y hacer clic con el botón izquierdo; luego desplazarse 10 cm a la izquierda en dirección paralela a la base del rectángulo y hacer clic con el botón derecho para soltar el chicle. Debió resultar un paralelogramo. Si quedó mal, utilizar *Edit-Delete* clic para borrar y repetir el procedimiento. Colorear el piso de amarillo.

La pared de la entrada a la planta se construye también con el *trapecio*, haciendo clic *izquierdo* en las esquinas de la izquierda de la pared posterior y en la izquierda del piso, luego subir 5 cm, hacer clic *izquierdo* y cerrar el paralelogramo con otro clic *izquierdo* y clic *derecho* para soltar el chicle. Si todo va bien, debe tener la perspectiva de dos paredes y del piso. Colorear con azul claro. Para construir las ventanas de la pared posterior, seleccionar el *rectángulo* y situar dos de éstos en la parte superior de la pared. Con el *fill* de la barra de dibujo, colorearlos de azul oscuro. La puerta de la entrada se hace con el *trapecio* como un paralelogramo; colorearla de azul oscuro. Seleccionar la primera ventana, ir a *Graphics-Fill Patterns-segundo cuadro hacia abajo* y hacer clic. La ventana debe quedar con una cuadrícula que semeja los vidrios. Haga lo mismo a la segunda ventana.

Para crear cuatro oficinas, necesita cinco divisiones de oficina. Utilizar el *trapecio* y dibujar un paralelogramo (1,80 x 2 cm), que debe colocar paralelo a la pared de entrada. Colorearlo de verde y, mediante *Graphics-Fill Patterns-último cuadro hacia abajo*, darle un efecto *dégradé*. Manteniendo seleccionado el paralelogramo (se deben ver cuatro cuadraditos en las esquinas), ir a *Edit-Copy* y luego a *Edit-Paste*. En la pantalla aparece un nuevo paralelogramo, al que puede arrastrar (con el botón izquierdo oprimido) y colocarlo paralelo a unos 2 cm del anterior. Se hace lo mismo con las otras tres divisiones.

En cada oficina se va a colocar un escritorio. Seleccionar el escritorio y colocarlo temporalmente en la parte izquierda del *Layout*. Lo puede agrandar arrastrando los cuadraditos de los lados; también lo puede girar con las herramientas que se encuentran casi debajo del *trapecio* y que parecen elipses con flecha. Cuando lo tenga en posición, ir a *Edit-Copy* y *Edit-Paste* para hacer los otros tres. Arrastrarlos para ponerlos en su sitio. Lo mismo debe hacer con el oficinista de camisa blanca.

A continuación se demarcan dos zonas en el piso: la de torneado y la de fresado. Con el *trapecio*, puede hacer el primer paralelogramo, colorearlo de rojo y darle efecto *dégradé* mediante *Edit-Copy*; luego, con *Edit-Paste* haga la otra zona y arrástrela para que coincida con el lado derecho-infe-

rior del piso. Colorearla de violeta. Seleccionar un torno (es el rojo, segunda fila, décima posición) y colocarlo en la zona roja de la izquierda. Repetir esta operación tres veces más. Los cuatro tornos deben quedar en la zona de torneado. Seleccione la fresa (es azul con blanco, segunda fila, novena posición) y colóquela en la otra zona, la de fresado. Ejecutar esta operación dos veces más. Seleccionar el operario (el de overol azul), agrandarlo un poco y, mediante *Edit-Copy* y luego *Edit-Paste*, hacer seis copias (¡aquí si está permitido clonar!) que debe arrastrar hasta el lado de cada una de las máquinas.

Para colocar el letrero, seleccionar la A de la barra de dibujo y hacer clic arriba de las ventanas de la pared posterior. En la pantalla aparece una ventana; escribir allí: "INDUSTRIAS su nombre", y pulsar *OK*.

Con el *trapecio*, construir el piso del parqueadero y colorearlo de gris. Con la *recta* (está debajo de la A) en la herramienta de dibujo, trazar tres rectas paralelas para simular las líneas de los parqueaderos. Mediante *Graphics-Line Color*, pintarlas de blanco. A continuación colocar un vehículo en cada parqueadero. Para colocar las estibas, baje una y, con *Edit-Copy* y luego *Edit-Paste*, elabore unas 20. También puede colocar tubos y cajas de cartón.

Para dar el efecto de que la puerta está abierta, construya con el *trapecio* una especie de corredor que conecte el parqueadero y la planta. Necesita dos figuras: un triángulo y un trapecio. Ahora, puede agregar más operarios, un montacargas con una estiba en la entrada y otro en el parqueadero.

Modelo 2. Entradas

En este modelo, como su nombre lo indica, las *Entidades Rueda amarilla, Rueda roja, Rueda azul y Rueda verde* llegan a la *Instalación Torno*, en el orden en que se les define. Las *Entidades* se identifican por medio de un *Atributo* y de la función *IF THEN ELSE*. Además, mediante una *Variable* y la función *INC*, se lleva la cuenta de las entradas de cada clase de *Entidad* al sistema.

Como siempre, comenzar haciendo clic en *File-New-Title* y escribir "Modelo-entradas-su nombre", en el renglón correspondiente al título. Verificar que la *librería* sea PROMODEL4.GLB, definir *Units* en minutos y *Distance* en metros. Salvar mediante *File-Save AS* en el disco (C:) en *Temp*, archivo "Entradas-su nombre" y hacer clic en *Save*. No olvidar salvar cada cinco minutos.

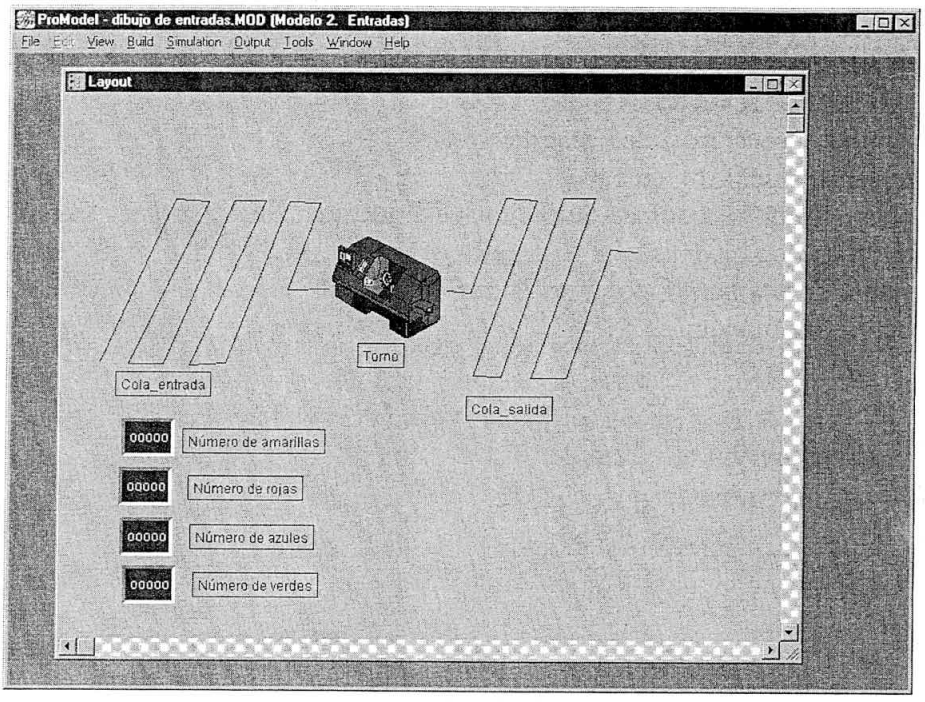


Figura 2.2 Layout del modelo con colas para entrada y salida.

1. *BUILD-Entities* (Seleccionar en la ventana *Entity Graphics* un círculo gris, que en la ventana *Name* lleva el nombre B*arrel Top,* editarlo con color amarillo, luego rojo, etc.)

Entities [5]

Icon	Name	Speed (mpm)	Stats...	Notes...
	Rueda_amarilla	50	Time Series	
	Rueda_roja	50	Time Series	
	Rueda_azul	50	Time Series	
	Rueda_verde	50	Time Series	

Figura 2.3 Entidades del modelo.

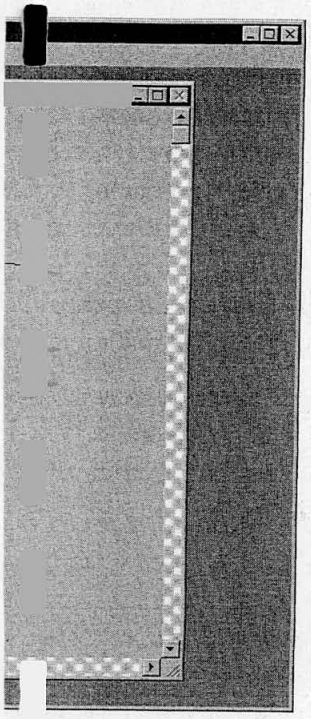
2. *BUILD (Locations)*

Locations [4]

Icon	Name	Cap.	Units	DTs...	Stats...	Rules...	Notes...
	Cola_entrada	100	1	None	Time Series	Oldest, FIFO	
	Torno	4	1	None	Time Series	Oldest	
	Cola_salida	100	1	None	Time Series	Oldest, FIFO	

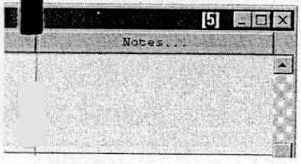
Figura 2.4 Instalaciones del modelo.

Para dibujar la *Instalación* Cola_entrada, ir a la ventana *Graphics* donde se encuentra un ícono que parece una escalera horizontal. Hacer clic sobre éste y luego sobre la ventana *Layout.* El primer clic *izquierdo* se hace en el sitio donde comienza la cola, los quiebres se hacen con clic *izquierdo*

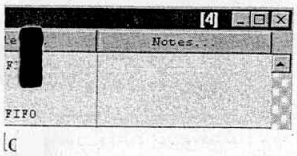


ntrada y salida.

a uics un círculo gris, *op*, editarlo con color



).



lc

ventana *Graphics* don-
l izontal. Hacer clic
r ic *izquierdo* se hace
ıcen con clic *izquierdo*

y al final se suelta el chicle con clic *derecho*. Verificar que la ventana *Locations* esté iluminada con el nombre *Loc1*. Borrarlo y escribir el nombre Cola_entrada. Sobre la gráfica de Cola_entrada, hacer *doble* clic. Aparece la ventana *Conveyor/Queue* donde se define la cola como *Queue* y como *Line*. Éste es un paso importante siempre que se definan colas. Si se omite, el programa asume que se trata de una banda transportadora (*Conveyor)* y puede causar un error que no deja iniciar el modelo. En esta ventana se puede definir la longitud de la cola en 100 m. La ventana se cierra con *OK*. Una flecha indica el sentido de la cola. Volver a la ventana *Locations*. En este caso, la capacidad de la cola se ha dejado como 100 *Entidades*.

El torno rojo se selecciona de manera similar. Se le cambia el nombre (verificar que esté iluminado *Lathe)* y la capacidad se define como 4. Esto quiere decir que pueden tornearse cuatro ruedas simultáneamente. La Cola_salida se hace de manera similar a la de entrada. Con la *A* de *File-Background-Front of the Grid,* se colocan los nombres.

3. *BUILD (Arrivals)*

Arrivals [5]

Entity...	Location...	Qty each...	First Time	Occurrences	Frequency	Logic	Disable
Rueda_amarilla	Cola_entrada	1	0	Infinite	1		No
Rueda_roja	Cola_entrada	2	0	Infinite	2		No
Rueda_azul	Cola_entrada	3	0	Infinite	3		No
Rueda_verde	Cola_entrada	4	0	Infinite	4		No

Figura 2.5 Llegadas del modelo.

La información del primer renglón se puede interpretar así: la *Entidad* Rueda_amarilla llega a la *Instalación* Cola_entrada de a 1. La primera vez no llega ninguna. Llegan infinitas ruedas con una *Frecuencia* de 1 cada minuto. De igual manera, la información del segundo renglón: La *Entidad* Rueda_roja llega a la *Instalación* Cola_entrada, de 2 en 2. La primera vez no llega ninguna. Llegan infinitas ruedas con una *Frecuencia* de 2 cada minuto, etc.

4. *BUILD (Attributes)*

Attributes [5]

ID	Type...	Classification...	Notes...
Amarillas	Integer	Ent	
Rojas	Integer	Ent	
Azules	Integer	Ent	
Verdes	Integer	Ent	

Figura 2.6 Atributos del modelo.

El *Atributo* es como una marca personal o una etiqueta que se pone, en este caso, a cada *Entidad*. Las Ruedas_amarillas se van a etiquetar con el *Atributo* "Amarillas", las rojas con "Rojas", etc.

5. *BUILD (Variables)*

Variables (global)

Icon	ID	Type	Initial value	Stats	Notes
Yes	Número_amarillas	Integer	0	Time Series	
Yes	Número_rojas	Integer	0	Time Series	
Yes	Número_azules	Integer	0	Time Series	
Yes	Número_verdes	Integer	0	Time Series	

Figura 2.7 Variables del modelo.

En la *Variable Entera* "Número_amarillas", se va a guardar la información del número de Ruedas amarillas que entran a la Cola_entrada; en la *Variable Entera* "Número_rojas", se va a guardar la información del número de Ruedas rojas que entran a la Cola_entrada; etc. El contador azul aparece al hacer clic en *Icon* y luego en el sitio donde se quiere colocar dentro de la ventana *Layout*. Para poner letreros, utilizar *File-Background-Front Of the Grid y* la letra *A*.

6. *BUILD (Processing)*

Entity	*Location*	*Operation*	*Output*	*Destination*	*Rules*
Rueda_amarilla	Cola_entrada	Amarillas = 1 **IF** Amarillas = 1 **THEN** **INC** Número_amarillas	Rueda_amarilla	Torno	First 1
Rueda_roja	Cola_entrada	Rojas= 1 **IF** Rojas = 1 **THEN** **INC** Número_rojas	Rueda_roja	Torno	First 1
Rueda_azul	Cola_entrada	Azules = 1 **IF** Azules = 1 **THEN** **INC** Número_azules	Rueda_azul	Torno	First 1
Rueda_verde	Cola_entrada	Verdes = 1 **IF** Verdes = 1 **THEN** **INC** Número_verdes	Rueda_verde	Torno	First 1
ALL	Torno	**WAIT** 1	ALL	Cola_salida	First 1
ALL	Cola_salida		ALL	EXIT	First 1

El *Proceso* de la *Entidad* Rueda_amarilla, es el siguiente: llega a la *Instalación* Cola_entrada y allí se realiza la siguiente *Operación:* Verificar si el *Atributo* Amarillas es 1; si lo es, entonces *incrementar* la *Variable* Número_amarillas en 1, si no, no hacer nada y la Rueda_amarilla sale hacia la *Instalación* Torno. Se hace lo mismo con las otras tres *Entidades*. En la quinta instrucción TODAS las *Entidades Esperan 1 minuto* (Tiempo de Operación del Torno) y siguen hacia la Cola_salida. En la sexta instrucción, TODAS las *Entidades* que están en Cola_salida salen hacia EXIT.

Para ejecutar paso a paso la *Operación* de la primera instrucción, hacer clic sobre *Operation*. Se abre una ventana donde hay un martillito. Hacer clic sobre el martillito y se abre otra ventana llamada *Logic Builder*. Oprimir el botón *Build Expression*. Como "Amarillas" es un *Atributo,* se busca en la ventana izquierda la palabra *Attributes*; al hacer clic sobre ella aparecen los cuatro *Atributos* definidos antes. Seleccionar "Amarillas" y colocar el cursor vertical después de la palabra subrayada en azul; hacer clic en el signo "=" y escribir "1"con el teclado o con *Keypad*. Pulsar el botón *Paste*. En la ventana del martillito debe aparecer "Amarillas = 1". Colocar el cursor debajo de "Amarillas = 1" y buscar en las *Funciones* **IF THEN ELSE.** Al seleccionarlo aparecen cuatro botones. Pulsar el botón *Build Expresión*. Buscar en *Attributes,* seleccionar "Amarillas" y escribir "=1". Pulsar *Paste*. Oprimir el botón *Statement* y seleccionar **INC.** Hacer clic en el botón *Identifier*. Aparecen las *Variables* definidas. Seleccionar "Número_amarillas". Pulsar el botón *Return* y luego *Paste*. En la ventana del martillito debe aparecer "**IF** Amarillas = 1 **THEN INC** Número_amarillas". Oprimir el botón *Close* para cerrar la ventana del martillito. Las otras instrucciones se editan de manera similar.

7. *SIMULATION (Options)*

En la pantalla aparece el botón denominado *Browser*. Al oprimirlo aparece la ventana *Examinar para encontrar carpeta*. Hacer un *doble* clic sobre el ícono del disco (C:), buscar el archivo *Temp* y oprimir *Aceptar*. En *Output path* debe aparecer " C:\Temp". Verificar que en el punto *Define run lenght by* aparezca *Time Only,* que el tiempo esté en minutos y los reportes, en *estándar*. Para definir el tiempo de simulación de 8 horas, escribir 8 en el renglón *Run Hours*. Cruzar los deditos y oprimir *Run*.

Si no se han cometido errores, el modelo debe comenzar a correr. Observe los valores que aparecen en los contadores y anótelos.

Modelo 3. Salidas

En este modelo, una *Entidad* llega a una *Instalación*, donde se procesa y se divide en varias partes. Cada subproducto sale por una banda transportadora diferente.

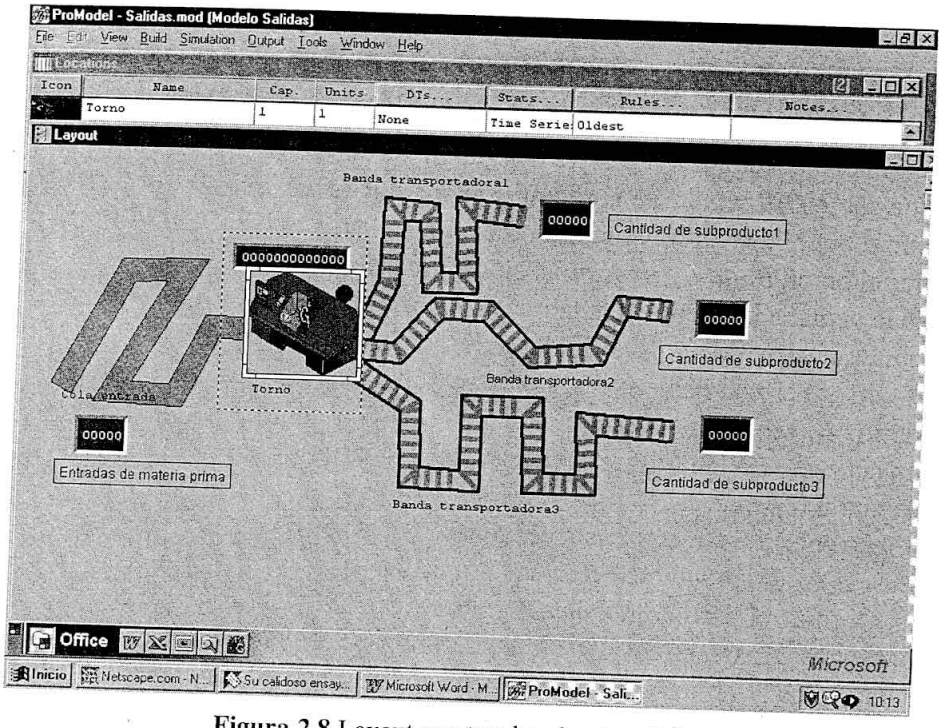


Figura 2.8 Layout con tres bandas de salida.

Comenzar, como siempre, haciendo clic en *File-New-Title*. Escribir "Modelo-Salidas-su nombre" en el renglón correspondiente al título. Verificar que la *librería* sea PROMODEL4.GLB, definir *Units* en minutos y *Distance* en metros. Salvar mediante *File-Save AS* en el disco (C:) en *Temp,* archivo "Salidas-su nombre" y hacer clic en *Save*. No olvidar salvar cada cinco minutos.

1. *BUILD (Entities)*

(Seleccionar en la ventana *Entity Graphics* una estiba cargada, que en la ventana *Entities* y en el renglón *Name* lleva el nombre de *Box_pallet,* que será la materia prima)

Icon	Name	Speed (mpm)	Stats...	Notes...
	Material	50	Time Series	
	Subprod	50	Time Series	

Figura 2.9 Entidades del modelo.

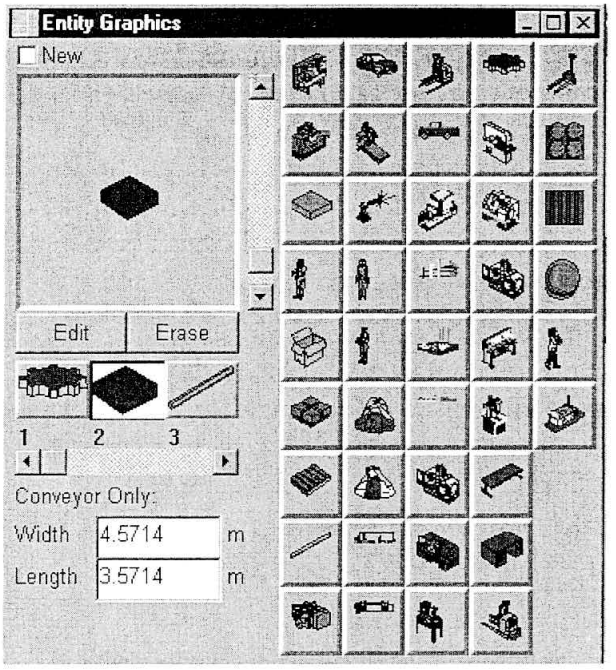


Figura 2.10 Gráficos de entidades.

En este modelo la *Entidad* Material, en forma de estiba, llega a la *Instalación* Torno, donde se divide en tres partes: subproducto 1, subproducto 2 y subproducto 3. Para tener un efecto gráfico de los tres subproductos, en el momento de dibujar la *Entidad* Subprod, en la ventana *Entity Graphics*, quite el chulo de *New*. Aparecen tres gráficas. Hacer clic en el botón de la gráfica 2 y luego en el ícono de lámina gris. Editar y cambiar de color. Hacer clic en el botón de la gráfica 3 y luego en el icono de barra o tubo. Editar y cambiar de color. Volver a colocar el chulo.

2. *BUILD (Locations)*

Icon	Name	Cap.	Units	DTs...	Stats...	Rules...	Notes...
	Cola_entrada	INFINIT	1	None	Time Series	Oldest, FIFO	
	Torno	1	1	None	Time Series	Oldest	
	Banda_transp1	INFINIT	1	None	Time Series	Oldest, FIFO	
	Banda_transp2	INFINIT	1	None	Time Series	Oldest, FIFO	
	Banda_transp3	INFINIT	1	None	Time Series	Oldest, FIFO	

Figura 2.11 Instalaciones del modelo.

Para observar la entrada de materia prima durante la simulación, se ha incluido una *Instalación* inicial en forma de cola. Sin embargo, esta Cola_entrada no es imprescindible. Definir la *Entidad* Cola_entrada como

una cola (*Queue)* y dibujarla con *Estilo* sólido. En este modelo se utiliza una nueva técnica para colocar los nombres. Hay tres ventanas abiertas: *Locations, Graphics y Layout.* En el renglón *Name* de la ventana *Locations*, escribir el nombre Cola_entrada. Ir a la ventana *Graphics* y quitar, con clic, un chulo (*check sign*) que allí aparece. Hacer clic dentro del rectángulo punteado que aparece rodeando a la cola; inmediatamente aparece el nombre. Para agrandar el letrero, hacer clic en el borde y arrastrar manteniendo el botón izquierdo oprimido. No olvidar volver a colocar el chulo.

Después, insertar la *Instalación* Torno. En el momento de colocar nombre (se ha quitado el chulo), hacer clic en la ventana *Graphics* sobre el botón **00** y luego clic dentro del rectángulo punteado alrededor del torno. Debe aparecer un contador. Después, hacer clic sobre el círculo azul y clic sobre una esquina del Torno. Debe aparecer un círculo azul dentro de un cuadrado. No olvidar colocar el chulo. En la ventana *Locations* colocar capacidad 3 para simular que en esa máquina, la materia prima se va a partir en tres.

A continuación, dibujar y colocar nombre a la primera banda transportadora. Hacer doble clic sobre la línea. Aparece la ventana *Conveyor/Queue*; señalar *Conveyor y Roller.* Dibujar a continuación las otras dos bandas, definiéndolas como *Conveyor y Roller.*

3. *BUILD (Arrivals)*

Entity...	Location...	Qty each...	First Time	Occurrences	Frequency	Logic	Disable
Material	Cola_entrada	1	0	100	1		No

Figura 2.12 Llegadas del modelo.

4. *BUILD (Variables)*

Icon	ID	Type...	Initial value	Stats...	Notes...
Yes	Cantidad_material	Integer	0	Time Series	
Yes	Num_subprod1	Integer	0	Time Series	
Yes	Num_subprod2	Integer	0	Time Series	
Yes	Num_subprod3	Integer	0	Time Series	

Figura 2.13 Variables del modelo.

e e modelo se utiliza
tres ventanas abiertas:
e la ventana *Locations*,
Graphics y quitar, con
ic dentro del rectángu-
d tamente aparece el
or y arrastrar mante-
ver a colocar el chulo.

n ito de colocar nom-
ana *Graphics* sobre el
lo alrededor del torno.
e l círculo azul y clic
culo azul dentro de un
ana *Locations* colocar
n eria prima se va a

i era banda transpor-
ta a *Conveyor/Queue*;
las otras dos bandas,

s	Frequency	Logic	Disable
	1		No

value	Stats...	Notes...
	Time Series	
	Time Series	
	Time Series	
	Time Series	

5. *BUILD (Processing)*

Entity	*Location*	*Operation*	*Output*	*Destination*	*Rule*
Material	Cola_entrada	**INC** Cantidad_material	Material	Torno	First 1
Material	Torno	**WAIT 2** **SPLIT 3 AS** subprod	Material	EXIT	First 1
Subprod	Torno		Subprod Subprod Subprod	Banda_transp1 Banda_transp3 Banda_transp3	Turn 1 Turn Turn
Subprod	Banda_transp1	**INC** Núm_subproductos1	Subprod	EXIT	First 1
Subprod	Banda_transp2	**INC** Núm_subproductos2 **GRAPHIC 2**	Subprod	EXIT	First 1
Subprod	Banda_transp3	**INC** Núm_subproductos3 **GRAPHIC 3**	Subprod	EXIT	First 1

Mediante la instrucción *SPLIT 3*, la *Entidad* Material se divide en tres *Entidades* Subprod. En la siguiente instrucción, en la parte del *Output*, aparecen 3 renglones. En el primero, al oprimir el botón *Rule*, se abre la ventana *Routing Rule;* seleccionar allí *By Turn*. Cuando termine de ingresar la información del primer renglón del *Output*, coloque el cursor vertical después de *Subprod* (el del *Output*) y oprima *Enter*. Aparece un renglón en blanco para llenar la información del segundo renglón. Hacer lo mismo para el tercer renglón.

El efecto de la instrucción *GRAPHIC* sobre la *Entidad* Subprod es cambiar el tipo de gráfica, para diferenciar los tres tipos de subproductos.

6. *SIMULATION (Options)*

En la pantalla aparece el botón denominado *Browser*. Al oprimirlo aparece la ventana *Examinar para encontrar carpeta*. Hacer *doble* clic sobre el ícono del disco (C:); buscar el archivo *Temp* y oprimir *Aceptar*. En *Output path* debe aparecer "C:\Temp". Verificar que en el punto *Define run lenght by* aparezca *Time Only,* que el tiempo esté en minutos y los reportes, en *estándar*.

Modelo 4. Entradas y salidas

En este modelo se combinan los conocimientos aprendidos en los dos modelos anteriores. Se trata de una *Instalación* a donde llegan múltiples productos que se procesan y salen por diferentes bandas transportadoras.

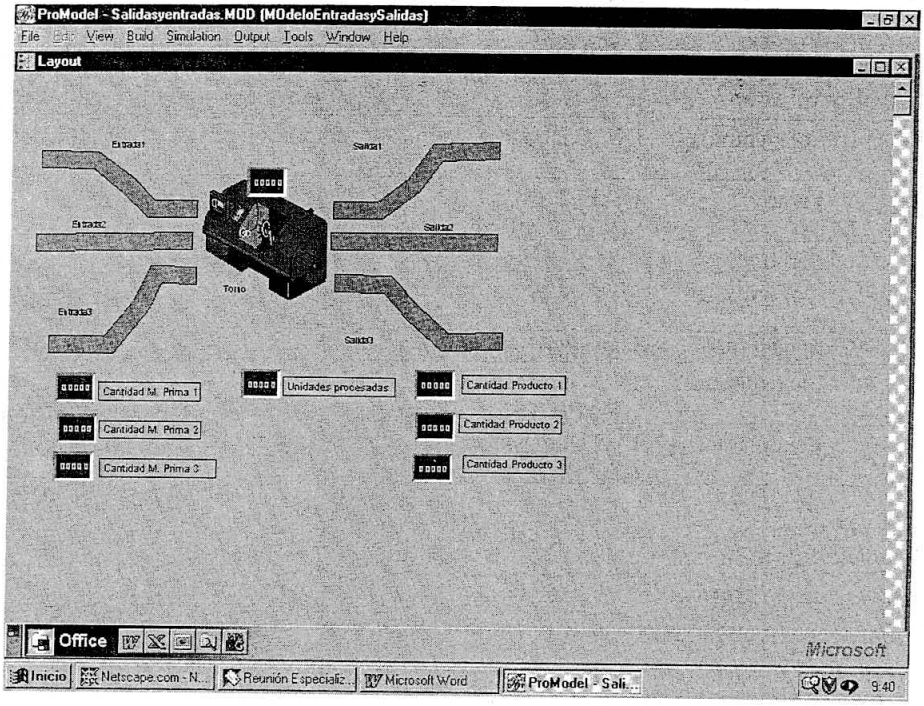


Figura 2.14 Layout del modelo con tres entradas y tres salidas.

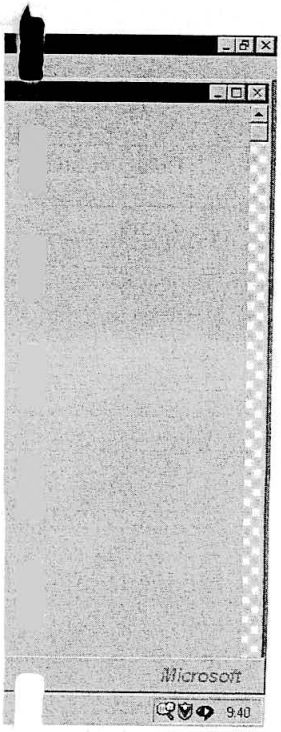
Comenzar, como siempre, haciendo clic en *File-New-Title*. Escribir "Modelo-Salidas-Entradas-su nombre", en el renglón correspondiente al título. Verificar que la *librería* sea PROMODEL4.GLB, definir *Units* en minutos y *Distance* en metros. Salvar mediante *File-Save AS* en el disco (C:), en *Temp,* archivo "Salidas-Entradas-su nombre" y hacer clic en *Save*. No olvidar salvar cada cinco minutos.

1. *BUILD (Entities)*

Entities [6]

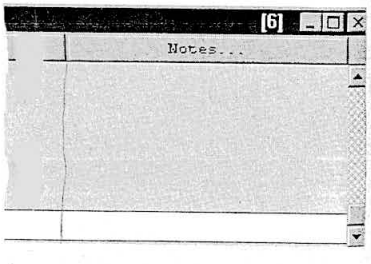
Icon	Name	Speed (mpm)	Stats...	Notes...
	Material1	50	Time Series	
	Material2	50	Time Series	
	Material3	50	Time Series	
	Producto1	50	Time Series	
	Producto2	50	Time Series	
	Producto3	50	Time Series	

Figura 2.15 Entidades del modelo.



y tres salidas.

...ew-Title. Escribir
n correspondiente al
l , definir *Units* en
-Save AS en el disco
y hacer clic en *Save*.



Seleccionar, en la ventana *Entity Graphics*, una estiba cargada que, en la ventana *Entities* y en el renglón *Name*, lleva el nombre de *Box_pallet*. Esta estiba será el material 1. *Pallet* será el material 2 y *PalletA* el material 3.

El material 1 llega a la entrada 1; el material 2 llega a la entrada 2 y el material 3 llega a la entrada 3.

2. *BUILD (Locations)*

Locations [7]

Icon	Name	Cap.	Units	DTs...	Stats...	Rules...	Notes...
	Entrada1	INFINIT	1	None	Time Series	Oldest, FIFO	
	Entrada2	INFINIT	1	None	Time Series	Oldest, FIFO	
	Entrada3	INFINIT	1	None	Time Series	Oldest, FIFO	
	Torno	3	1	None	Time Series	Oldest	
	Salida1	INFINIT	1	None	Time Series	Oldest, FIFO	
	Salida2	INFINIT	1	None	Time Series	Oldest, FIFO	
	Salida3	INFINIT	1	None	Time Series	Oldest, FIFO	

Figura 2.16 Instalaciones del modelo.

Definir las entradas y salidas como colas (*Queues*), y dibujarlas con *Estilo* sólido.

3. *BUILD (Arrivals)*

Arrivals [3]

Entity...	Location...	Qty each...	First Time	Occurrences	Frequency	Logic	Disable
Material1	Entrada1	1	0	100	1		No
Material2	Entrada2	1	0	100	1		No
Material3	Entrada3	1	0	100	1		No

Figura 2.17 Llegadas del modelo.

4. *BUILD (Variables)*

Variables (global) [8]

Icon	ID	Type...	Initial value	Stats...	Notes...
Yes	Cant_material1	Integer	0	Time Series	
Yes	Cant_material2	Integer	0	Time Series	
Yes	Cant_material3	Integer	0	Time Series	
Yes	U_procesadas	Integer	0	Time Series	
Yes	Num_producto1	Integer	0	Time Series	
Yes	Num_producto2	Integer	0	Time Series	
Yes	Num_producto3	Integer	0	Time Series	

Figura 2.18 Variables del modelo.

5. *BUILD (Processing)*

Entity	*Location*	*Operation*	*Output*	*Destination*	*Rule*
Material1	Entrada1	**INC** Cant_material1	Material1	Torno	First
Material2	Entrada2	**INC** Cant_material2	Material2	Torno	First
Material3	Entrada3	**INC** Cant_material3	Material3	Torno	First
Material1	Torno	**WAIT 1** **INC** U_procesadas	Material1	Salida1	First
Material2	Torno	**WAIT 1** **INC** U_procesadas	Material2	Salida2	First
Material3	Torno	**WAIT 1** **INC** U_procesadas	Material3	Salida3	First
Material1	Salida1	**RENAME AS** Producto1	Material1	EXIT	First
Material2	Salida2	**RENAME AS** Producto2	Material2	EXIT	First
Material3	Salida3	**RENAME AS** Producto3	Material3	EXIT	First
Producto1	Salida1	**INC** Núm_producto1	Producto1	EXIT	First
Producto2	Salida2	**INC** Núm_producto2	Producto2	EXIT	First
Producto3	Salida3	**INC** Núm_producto3	Producto3	EXIT	First

Observar que cuando se utiliza la instrucción *RENAME* en el *Output*, la entidad vieja debe salir del sistema, mediante *Exit*.

6. *SIMULATION (Options)*

En la pantalla aparece el botón llamado *Browser*. Al oprimirlo aparece la ventana *Examinar para encontrar carpeta*. Hacer doble clic sobre el ícono del disco (C:), buscar el archivo *Temp* y oprimir *Aceptar*. En *Output path* debe aparecer "C:\Temp". Verificar que en el punto *Define run lenght by* aparezca *Time Only,* que el tiempo esté en minutos y los reportes, en *estándar*. Oprimir el botón *Run*.

Modelo 5. Máquinas

En este modelo, como su nombre lo indica, se revisan algunas formas para programar mantenimiento preventivo en las máquinas cada determinado tiempo o cada que se elabore determinado número de piezas. También se pueden programar las pausas de descanso.

Output	Destination	Rule
[a ial1	Torno	First 1
[aterial2	Torno	First 1
[a ial3	Torno	First 1
a ial1	Salida1	First 1
aterial2	Salida2	First 1
aterial3	Salida3	First 1
a ial1	EXIT	First 1
a ial2	EXIT	First 1
aterial3	EXIT	First 1
o cto1	EXIT	First 1
o cto2	EXIT	First 1
oducto3	EXIT	First 1

V. ME en el *Output*, la

l primirlo aparece la
ble clic sobre el ícono
ir. En *Output path*
fine run lenght by
os y los reportes, en

algunas formas para
cada determinado
piezas. También se

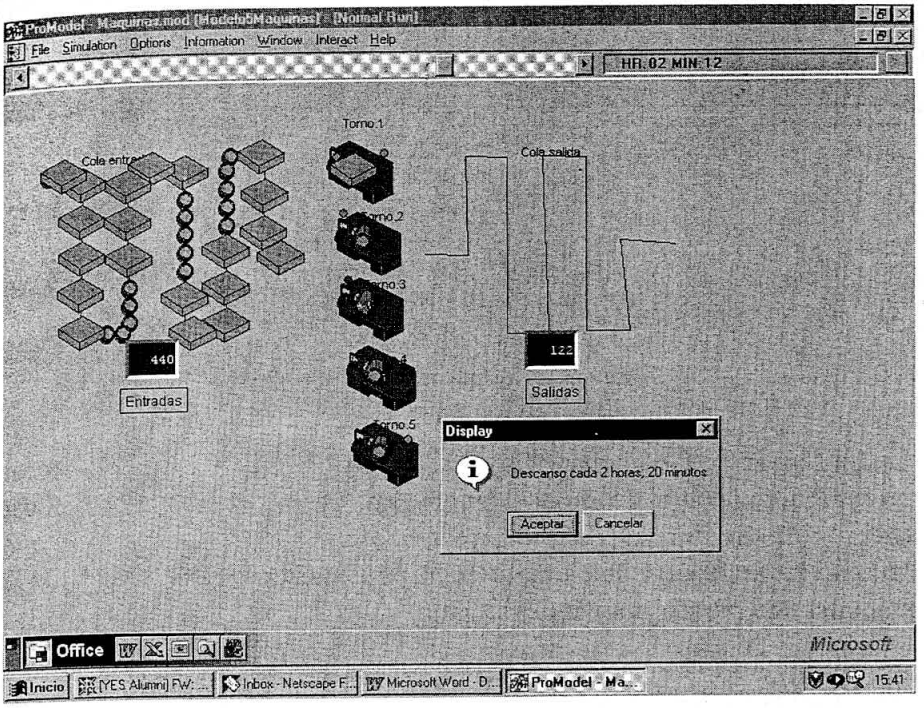


Figura 2.19 Layout con cinco máquinas, una cola de entrada y otra de salida.

Comenzar, como siempre, haciendo clic en *File-New-Title*. Escribir "Modelo-Máquinas-su nombre" en el renglón correspondiente al título. Verificar que la *librería* sea PROMODEL4.GLB, definir *Units* en minutos y *Distance* en metros. Salvar mediante *File-Save AS* en el disco (C:), en *Temp,* archivo "Máquinas-su nombre" y hacer clic en *Save*. No olvidar salvar cada cinco minutos.

7. *BUILD (Entities)*

Entities [2]

Icon	Name	Speed (mpm)	Stats...	Notes...
	Esfera	50	Time Series	
	Prisma	50	Time Series	

Figura 2.20 Entidades del modelo.

8. BUILD (Locations)

Locations [9]

Icon	Name	Cap.	Units	DTs	Stats	Rules	Notes
	Cola_entrada	INFINIT	1	None	Time Series	Oldest, FIFO	
	Cola_salida	INFINIT	1	None	Time Series	Oldest, FIFO	
	Torno	1	5	None	Time Series	Oldest, First	
	Torno.1	1	1	Setup,	Time Series	Oldest	
	Torno.2	1	1	Usage,	Time Series	Oldest	
	Torno.3	1	1	Clock,	Time Series	Oldest	
	Torno.4	1	1	Entry,	Time Series	Oldest	
	Torno.5	1	1	None	Time Series	Oldest	

Figura 2.21 Instalaciones del modelo.

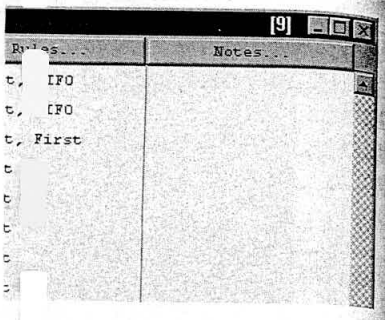
En el momento de poner el nombre "Torno", quitar el chulo y colocar un círculo azul sobre el torno y el letrero con el botón **Aa**. Volver a colocar el chulo. El círculo azul, en la simulación, es una especie de semáforo que está en verde cuando la máquina se encuentra en operación; rojo, cuando la máquina está parada por mantenimiento, y azul, cuando no está ocupada.

En el momento de asignar el número 5 a las unidades del Torno, aparecen 5 tornos en posición horizontal. Éstos se deben arrastrar para ponerlos en posición vertical. Si omitió la operación anterior, la puede hacer con cada uno individualmente.

DOWNTIMES: Intervalos de tiempo en los que la máquina permanece inactiva. ProModel® provee cuatro maneras para detener las máquinas. A) *Set Up:* tiempo que se emplea para alistar la producción cada vez que se cambia de producto; también se conoce como tiempo de *aprestamiento o alistamiento*. B) *Usage*: mantenimiento que se programa después que la máquina ha producido un número determinado de piezas. C) *Clock:* paradas de máquina que se programan de acuerdo con un horario. Con esta opción se pueden programar las pausas de descanso de los operarios cuando las máquinas no son automáticas. D) *Entry:* paradas de mantenimiento que se programan para la máquina después de determinado tiempo de trabajo; por ejemplo, la lubricación. Instrucciones específicas se pueden incluir dentro de la *Lógica*.

(*) *Set Up*

Entity	*Prior Entity*	*Logic*	*Disable*
Esfera	Prisma	**WAIT** 5 **DISPLAY** "Torno 1. Set Up para Esfera"	No
Prisma	Esfera	**WAIT** 5 **DISPLAY** "Torno1. Set Up para Prisma"	No



	Disable
	No
	No

Con esta instrucción, cada vez que se cambia de *Entidad*, se emplea medio minuto para hacer el alistamiento. Durante la simulación, aparece un aviso indicando que la pausa es por *Set Up*.

(**) *Usage*

Frequency	*First Time*	*Priority*	*Logic*	*Disable*
400	400	99	**DISPLAY** " Torno2. Mantenimiento por uso cada 400" **WAIT** 5	No

Con esta instrucción, la máquina se detiene 5 minutos para mantenimiento, después de haber producido 400 piezas. La opción de prioridad se utiliza cuando hay empate en el número de piezas producidas y se quiere hacer mantenimiento a determinada máquina. A ésta se le da mayor prioridad.

(***) *Clock*

Frequency	*First Time*	*Priority*	*Scheduled*	*Logic*	*Disable*
120	120	99	No	**DISPLAY** "Torno 3. Descanso de 20 minutos cada dos horas" **WAIT** 20	No

Con esta instrucción, la máquina para cada dos horas durante 20 minutos.

(****) *Entry*

Frequency	*First Occurrence*	*Logic*	*Disable*
100	100	**DISPLAY** "Torno 4.Lubricación después de 100 piezas" **WAIT** 7	

Con esta instrucción, la máquina se detiene 7 minutos para hacerle lubricación después de elaborar 100 piezas.

9. *BUILD (Arrivals)*

Arrivals [3]

Entity...	Location...	Qty each...	First Time	Occurrences	Frequency	Logic	Disable
Esfera	Cola_entrada	5; Llegdas	0	1			No
Prisma	Cola_entrada	5; Llegdas	0	1			No

Figura 2.22 Llegadas del modelo.

En este caso las dos *Entidades* llegan en grupos de a cinco. Observar que no se ha definido la *Frecuencia*, porque esta se va a determinar por medio de un *Ciclo de Llegadas*, denominado "Llegadas".

10. *BUILD (More Elements-Arrival Cycles)*

Arrival Cycles [2]

ID	Qty / %	Cumulative...	Table...
Llegdas	Quantity	No	Defined

Figura 2.23 Ciclo de llegadas del modelo.

En una tabla, el *Ciclo de Llegadas* indica la cantidad de grupos de a cinco que deben llegar al sistema durante las ocho horas de simulación.

TABLE For Llegadas

Table for Llegdas [8]

Time (Hours)	Qty / %
1	5
2	10
3	2
4	5
5	2
6	10
7	5
8	5

Figura 2.24 Tabla de tiempos del ciclo de llegadas del modelo.

Volver a *BUILD-Arrivals-Quantity Each*, donde se abre la ventana *Arrival Qty*. Allí debe aparecer el letrero "Llegadas"; señalarlo y dar *OK*.

11. *BUILD (Variables)*

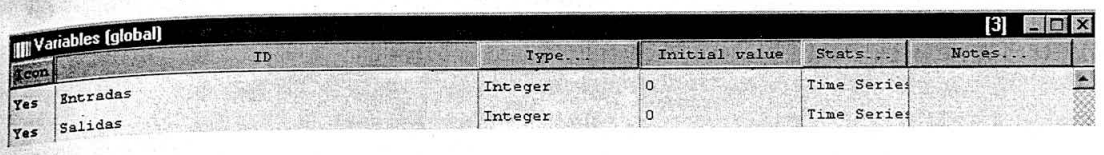


Figura 2.25 Variables del modelo.

12. *BUILD (Processing)*

Entity	*Location*	*Operation*	*Output*	*Destination*	*Rule*
ALL	Cola_entrada	**INC** Entradas	ALL	Torno	First 1
ALL	Torno	**WAIT 5**	ALL	Cola_salida	First 1
ALL	Cola_salida	**INC** Salidas	ALL	EXIT	First 1

13. *SIMULATION (Options)*

En la pantalla aparece el botón llamado *Browser*. Al oprimirlo aparece la ventana *Examinar para encontrar carpeta*. Hacer un *doble* clic sobre el ícono del disco (C:), buscar el archivo *Temp* y oprimir *Aceptar*. En *Output path* debe aparecer "C:\Temp". Verificar que en el punto *Define run lenght by* aparezca *Time Only*, que el tiempo esté en minutos y los reportes, en *estándar*. Definir *Run Hours* como 8. Oprimir el botón *Run*.

Modelo 6. Variabilidad en entradas y máquinas

En los modelos, vistos hasta ahora, los tiempos de llegada de los materiales al sistema han sido *discretos*, esto es, conocidos y constantes. En este modelo se incluye *variabilidad* en las entradas y en el proceso.

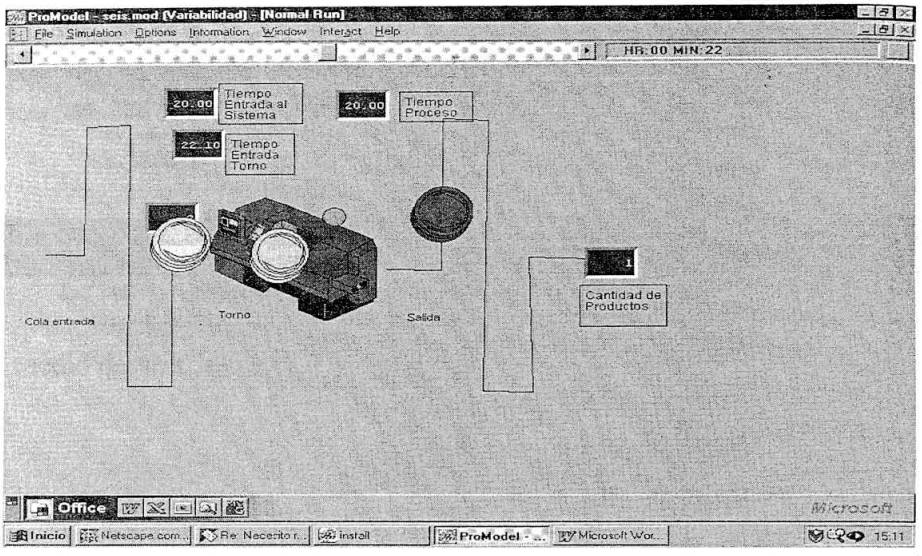


Figura 2.26 Layout del modelo.

Comenzar, como siempre, haciendo clic en *File-New-Title*. Escribir "Modelo-Variabilidad-su nombre" en el renglón correspondiente al título. Verificar que la *librería* sea PROMODEL4.GLB, definir *Units* en minutos y *Distance* en metros. Salvar mediante *File-Save AS* en el disco (C:), en *Temp,* archivo "Variabilidad-su nombre" y hacer clic en *Save*. No olvidar salvar cada cinco minutos.

1. *BUILD (Entities)*

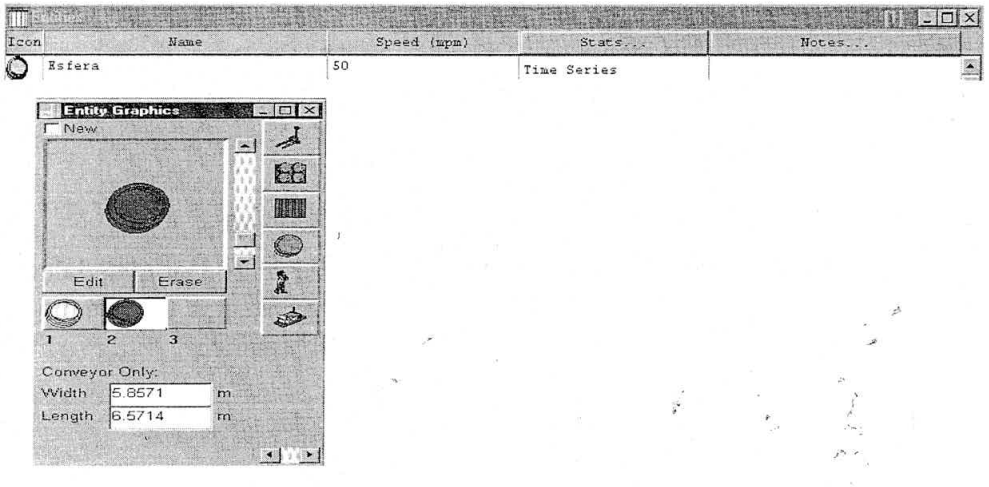
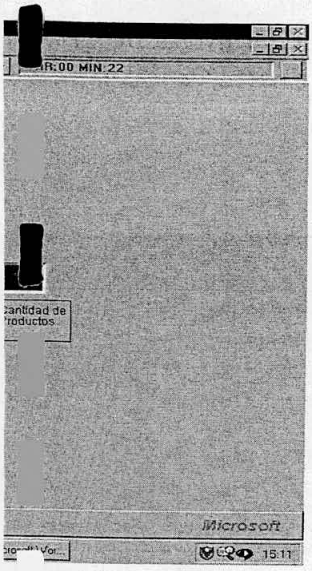


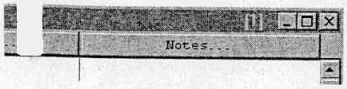
Figura 2.27 Entidad del modelo con dos gráficos.



2. *BUILD (Locations)*

Locations [4]

Icon	Name	Cap.	Units	DTs...	Stats...	Rules...	Notes...
	Cola_entrada	INFINIT	1	None	Time Series	Oldest, FIFO	
	Salida	INFINIT	1	None	Time Series	Oldest, FIFO	
	Torno	1	1	Clock, Entry,	Time Series	Oldest	



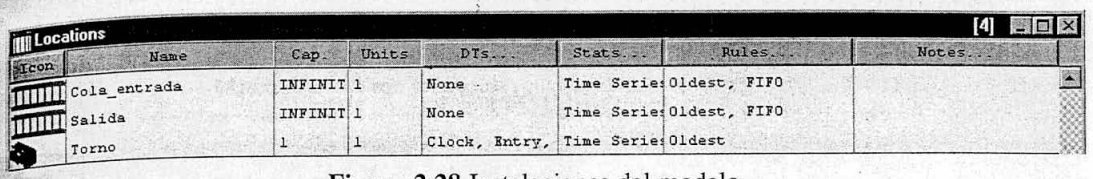


Figura 2.28 Instalaciones del modelo.

Definir la Cola Entrada como (*Queue*) y la Salida como (*Conveyor*), y dibujarlas con *Estilo* line.

DOWNTIMES. A) *Clock:* paradas de máquina, que se programan de acuerdo con un horario. Con esta opción se pueden programar las pausas de descanso de los operarios cuando las máquinas no son automáticas. B) *Entry:* paradas de mantenimiento que se programan para la máquina después de determinado tiempo de trabajo; por ejemplo, la lubricación. Instrucciones específicas se pueden incluir dentro de la *Lógica*.

DTs-Clock

Frequency	*First Time*	*Priority*	*Scheduled*	*Logic*	*Disable*
240	240	99	No	**DISPLAY** "Pausa de descanso" **WAIT** 15	No

Con esta instrucción, la máquina hace una pausa de 15 minutos cada dos horas.

DTs-Entry

Frequency	*First Occurrence*	*Logic*	*Disable*
30	30	**DISPLAY** "Lubricación después de 30 piezas" **WAIT** 5	

Con esta instrucción, la máquina se detiene 5 minutos para hacer lubricación después de elaborar 30 piezas.

3. *BUILD (Arrivals)*

Arrivals [2]

Entity...	Location...	Qty each...	First Time	Occurrences	Frequency	Logic	Disable
Esfera	Cola_entrada	1	0	10	10		No

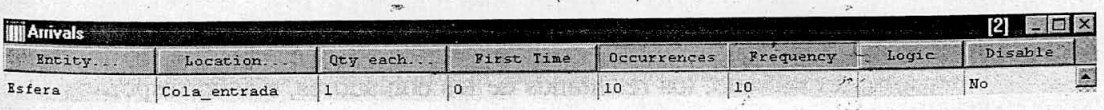


Figura 2.29 Llegadas del modelo.

En este caso llegan al sistema 10 esferas (*Ocurrences)*, una (*Qty. Each)* cada 10 minutos (*Frequency).*

4. *BUILD (Variables)*

Variables (global) [6]

Icon	ID	Type...	Initial value	Stats...	Notes...
Yes	Num_entradas	Integer	0	Time Series	
Yes	Ti_entrada	Real	0	Time Series	
Yes	Ti_torno	Real	0	Time Series	
Yes	Cant_prod	Integer	0	Time Series	
Yes	Ti_en_torno	Real	0	Time Series	

Figura 2.30 Variables del modelo.

En la *Variable* Ti_entrada, se guarda el tiempo en minutos en que entra cada Esfera al sistema. En la *Variable* Ti_en_torno, el tiempo en que entra cada Esfera al Torno se guarda en minutos y en la *Variable* Ti_torno, el tiempo de procesamiento de la Esfera en el Torno se guarda en minutos.

5. *BUILD (Processing)*

Entity	*Location*	*Operation*	*Output*	*Destination*	*Rule*
Esfera	Cola_entrada	Ti_entrada = **CLOCK(MIN)** **INC** Num_entradas	Esfera	Torno	First 1
Esfera	Torno	Ti_en_torno = **CLOCK(MIN)** **WAIT** 20 **GRAPHIC** 2	Esfera	Salida	First 1
Esfera	Salida	Ti_torno = **CLOCK(MIN)**-Ti_en_torno **INC** Cant_prod	Esfera	EXIT	First 1

6. *SIMULATION (Options)*

Aparece el botón denominado *Browser*. Al oprimirlo aparece la ventana *Examinar para encontrar carpeta*. Hacer un *doble* clic sobre el ícono del disco (C:). Buscar el archivo *Temp* y oprimir *Aceptar*. En *Output path* debe aparecer "C:\Temp". Verificar que en el punto *Define run lenght by*, aparezca *Time Only,* que el tiempo esté en minutos y los reportes, en *estándar*. Oprimir el botón *Run*.

Al correr el Modelo, los resultados de los contadores, que corresponden a la décima Esfera, son los siguientes:

VARIABLES

Variable Name	Total Changes	Average Minutes Per Change	Minimum Value	Maximum Value	Current Value	Average Value
Num entradas	10	9.00	0	10	10	7.78
Ti entrada	10	9.00	0	90	90	67.89
Ti torno	10	20.20	0	20	20	17.83
Cant prod	10	20.20	0	10	10	4.49
Ti en torno	10	18.20	0	182	182	91.78

Tabla 2.1 Resultados de la simulación del modelo.

Ahora, en la ventana *Simulation Complete* oprimir el botón *SI*. Oprimir el octavo botón de izquierda a derecha (es el que tiene tres líneas entrecuzadas).

Aparece la ventana *Choose Statistics for Plot*. En la parte izquierda debe aparecer el letrero: *Cola_entrada Contents*. Seleccionarlo. En la mitad, entre las dos ventanitas, hay una flecha indicando hacia la derecha; hacer clic sobre ésta. El nombre *Cola_entrada Contents* debió pasar a la ventanita derecha. De la misma manera seleccionar *Ti_en_torno Value History* y *Ti_torno Value History*. Dar *OK*. Aparece una gráfica con colores, donde se puede observar *Ti_en_torno Value History* como una función escalonada creciente con escalones de altura 20 (el tiempo de proceso del torno), y *Ti_torno Value History* como una función constante con altura 20.

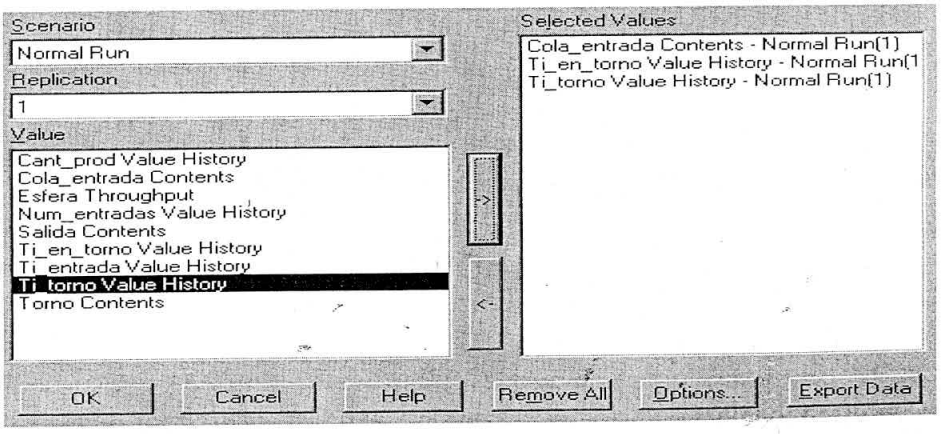


Figura 2.31 Ventana para seleccionar las variables del modelo que se quieren analizar.

7. Introducir variabilidad

Para introducir *Variabilidad* en las *Llegadas*, se cambia por N(10,5) el número 10 que aparece en *BUILD-Arrivals-Frequency*. Este cambio significa que las Esferas van a llegar ahora con una Distribución Normal de media 5 y desviación estándar 5.

Para introducir *Variabilidad en el Tiempo de Operación del Torno*, se cambia *WAIT 20* por *WAIT N(20,10)* en *BUILD-Processing*, en el segundo renglón con la Esfera en Torno. Este cambio significa que la operación en el torno puede demorarse entre 10 y 30 minutos.

Ahora ir a *SIMULATION* y dar *Run*. Los resultados son los siguientes:

VARIABLES

Variable Name	Total Changes	Average Minutes Per Change	Minimum Value	Maximum Value	Current Value	Average Value
Num entradas	10	8.19	0	10	10	8.39
Ti entrada	10	8.19	0	81.97	81.97	62.61
Ti torno	10	20.16	0	36.46	30.67	16.35
Cant prod	10	20.16	0	10	10	4.28
Ti en torno	10	17.09	0	170.96	170.96	88.22

Tabla 2.2 Resultados con variabilidad.

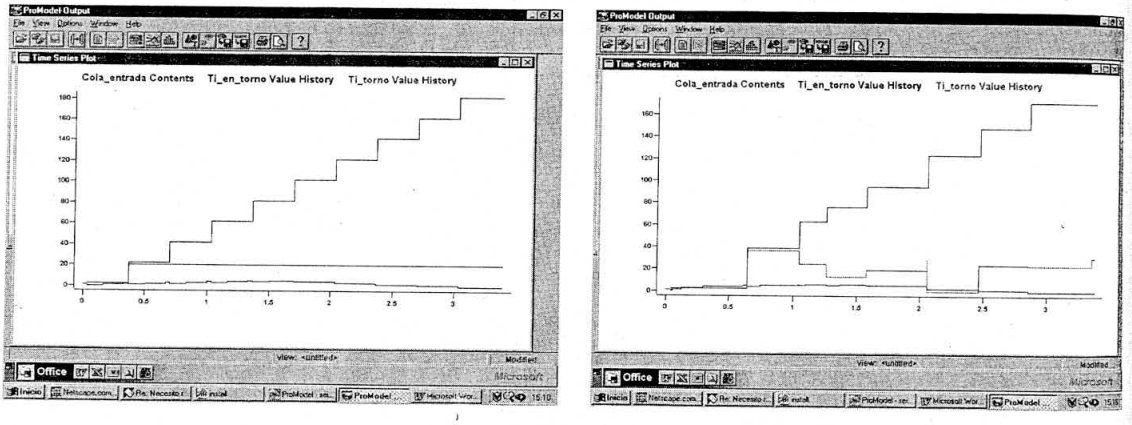


Figura 2.32 A la izquierda, resultados sin variabilidad; a la derecha, resultados con variabilidad.

Modelo 7. Rutas y recursos

Con este modelo, se ilustra el uso de *Trayectorias o Rutas* por donde se desplazan los *Recursos* que pueden ser humanos o físicos.

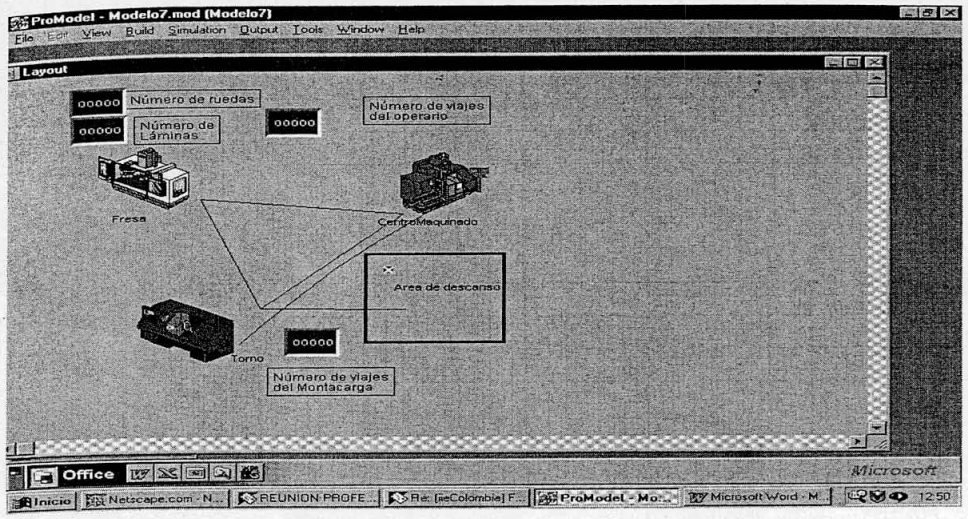


Figura 2.33 Layout del modelo con dos rutas: una triangular y una recta.

Comenzar, como siempre, haciendo clic en *File-New-Title*. Escribir "Modelo-Rutas-su nombre" en el renglón correspondiente al título. Verificar que la *librería* sea PROMODEL4.GLB, definir *Units* en minutos y *Distance* en metros. Salvar mediante *File-Save AS* en el disco (C:), en *Temp,* archivo "Rutas-su nombre" y hacer clic en *Save*. No olvidar salvar cada cinco minutos.

1. *BUILD (Entities)*

Icon	*Name*	*Speed (m.p.m)*	*Statistics*
Esfera (Barrel-Top)	Rueda	50	Time Series
Lámina (Raw-Material)	Lámina	50	Time Series

2. *BUILD (Locations)*

Icon	*Name*	*Cap.*	*Units*	*Dts*	*Stats*	*Rules*
Fresadora (Mill)	Fresa	2	1	None	Time Series	Oldest
Torno rojo (Lathe)	Torno	2	1	None	Time Series	Oldest
C. de Mecanizado	CentroMecanizado	2	1	None	Time Series	Oldest
	Area de descanso	1	1	None	Time Series	Oldest

3. BUILD (Arrivals)

Entity	Location	Qty. Each	First Time	Occurr.	Freq.	Disable
Rueda	Fresa	1	0	500	2	No
Lámina	Fresa	1	0	500	1	No

4. BUILD (Attributes)

Identification	Type	Classification	Notes
Ruedas	Integer	Ent.	
Láminas	Integer	Ent.	

5. BUILD (Variables)

Icon	Identification	Type	Initial Value	Statistics
Yes	Num_viajes_operario	Integer	0	Time Series
Yes	Num_viajes_montacargas	Integer	0	Time Series
Yes	Num_ruedas	Integer	0	Time Series
Yes	Num_láminas	Integer	0	Time Series

6. BUILD (Path Networks)

Graphic	Name	Type	T/S	Paths	Interfaces	Mapping	Nodes
	Ruta Triangular	Passing	Speed&D	4	4	0	4
	Ruta Montacargas	Passing	Speed&D	1	2	0	2

Ruta Triangular *Paths* *Interfaces.*

From	To	Bi	Distance		Nodes	Location
N1	N2	Bi	25		N1	Fresa
N2	N4	Bi	25		N2	CentroMaquinado
N4	N2	Bi	20		N4	Torno
N4	N3	Bi	18		N3	Area_de_Descanso

Ruta Montacargas *Paths* *Interfaces.*

From	To	Bi	Distance		Nodes	Location
N1	N2	Bi	36		N1	CentroMaquinado
					N2	Torno

Time	Occurr.	Freq.	Disable
	500	2	No
	500	1	No

ation	Notes

In	al Value	Statistics
	0	Time Series
	0	Time Series
	0	Time Series
	0	Time Series

n	faces	Mapping	Nodes
	4	0	4
	2	0	2

's.

	Location
	Fresa
	CentroMaquinado
	Torno
	Area de Descanso

	Location
	CentroMaquinado
	Torno

7. *BUILD (Resources)*

Icon	Name	Units	DTS	Stats	Specs	Search	Logic	Notes
	Operario	1	Clock	By Unit	Ruta Triangular	None	0	
	Montacargas	1	Clock	By Unit	Ruta Montacarga	None	0	

DTs- Clock Downtimes for Operario

Frequency	First Time	Priori	Sched	List	Node	Logic	Disable
120	120		No		N3	**DISPLAY** "Pausa de descanso" **WAIT** 20	No

Para definir *Specifications* del *Recurso* Operario, oprimir el botón. Aparece una ventana con este nombre. Allí se seleccionan las siguientes alternativas: *Path Network*: Ruta Triangular, *Home*: N3, *Return Home if Idle*: colocar el chulo, para que cada vez que el operario esté desocupado vaya a su sitio en la *Instalación* Área de Descanso. *Off Shift*: N3, *Break*: N3. Con estas selecciones, el Operario debe ir al Área de Descanso, cuando no esté en turno o cuando haga pausa.

DTs- Clock Downtimes for Montacargas

Frequency	First Time	Priori	Sched	List	Node	Logic	Disable
120	120		No		N1	**DISPLAY** "Pausa para revisión" **WAIT** 5	No

Para definir *Specifications* del Recurso Montacargas, oprimir el botón y aparece una ventana con este nombre. Allí se seleccionan las siguientes alternativas: *Path Network*: Ruta Montacargas, *Home*: N1, *Return Home if Idle*: Colocar el chulo, para que cada vez que el Montacargas esté desocupado vaya a su sitio en la *Instalación* Centro Maquinado. *Off Shift*: N1; *Break*: N1. Con estas selecciones, el Montacargas debe ir al Centro de Maquinado, cuando no esté en actividad o cuando haga pausa.

8. *BUILD (Processing)*

Entity	*Loc.*	*Operation*	*Output*	*Dest.*	*Rul*	*Logic*
Rueda	Fresa	Ruedas = 1 **IF** Ruedas =1 **THEN** {**INC** Núm_ruedas}	Rueda	Torno	First 1	**MOVE WITH** Operario **THEN FREE** **INC** Num_viajes_operario
Lámina	Fresa	Láminas = 1 **IF** Láminas =1 **THEN** {**INC** Núm_láminas}	Lámina	Torno	First 1	**MOVE WITH** Operario **THEN FREE** **INC** Num_viajes_operario
Rueda	Torno	**GET** 1 Montacargas	Rueda	Centro Maquinado	First 1	**MOVE WITH** Montacargas **THEN FREE** **INC** Num_viajes_montacargas
Lámina	Torno	**GET** 1 Montacargas	Lámina	Centro Maquinado	First 1	**MOVE WITH** Montacargas **THEN FREE** **INC** Num_viajes_montacargas
Rueda	Centro Maquinado	**FREE ALL**	Rueda	EXIT	First 1	
Lámina	Centro Maquinado	**FREE ALL**	Lámina	EXIT	First 1	

9. *SIMULATION (Options)*

En la pantalla aparece el botón llamado *Browser*. Al oprimirlo aparece la ventana *Examinar para encontrar carpeta*. Hacer *doble* clic sobre el ícono del disco (C:). Buscar el archivo *Temp* y oprimir *Aceptar*. En *Output path* debe aparecer "C:\Temp". Verificar que en el punto *Define run lenght by* aparezca *Time Only*, que el tiempo esté en minutos y los reportes, en *estándar*. Oprimir el botón *Run*.

Modelo 8. Operarios y turnos

Con este modelo, se ilustra el uso de *Turnos* para programar los *Recursos*, que pueden ser humanos o físicos. Se definen dos *Recursos*, cada uno de los cuales trabaja en un turno. Comenzar, como siempre, haciendo clic en *File-New-Title*. Escribir "Modelo-Turnos-su nombre" en el renglón correspondiente al título. Aquí se incluye un pequeño cambio. Verificar que la *Librería* sea *PEOPLE.GLB*, definir *Units* en minutos y *Distance* en metros. Salvar mediante *File-Save AS* en el disco (C:), en *Temp*, archivo "Turnos-su nombre" y hacer clic en *Save*. No olvidar salvar cada cinco minutos.

ul	Logic
irs	**MOVE WITH** Operario **THEN FREE**
	INC Num_viajes_operario
irst	**MOVE WITH** Operario **THEN FREE**
	INC Num_viajes_operario
rst	**MOVE WITH** Montacargas **THEN FREE**
	INC Num_viajes_montacargas
rst	**MOVE WITH** Montacargas **THEN FREE**
	INC Num_viajes_montacargas
rst	
rsi	

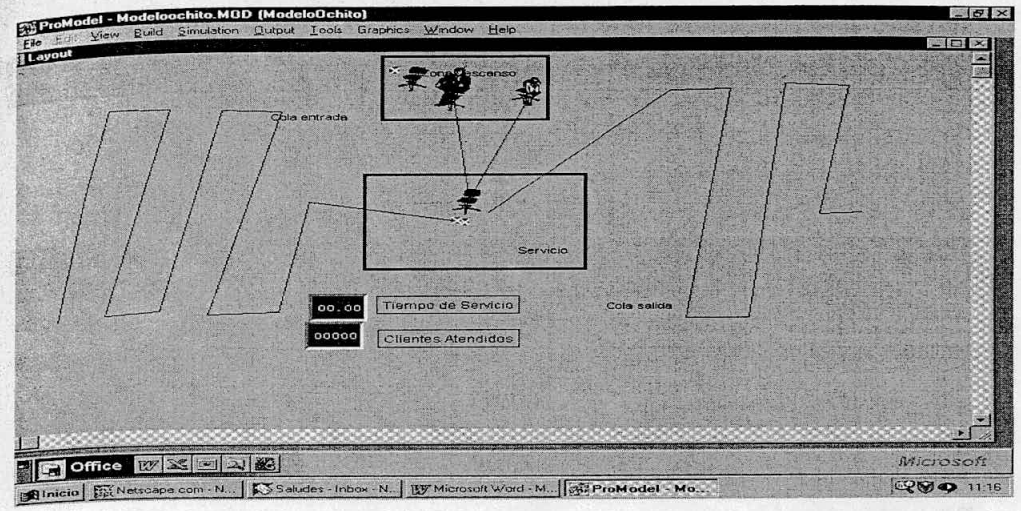


Figura 2.34 Layout con zona de descanso para los operarios.

1. *BUILD (Entities)*

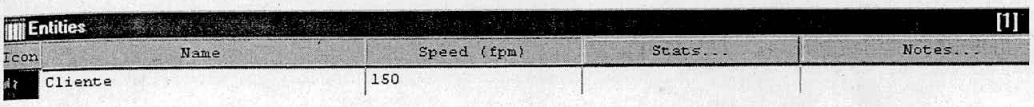
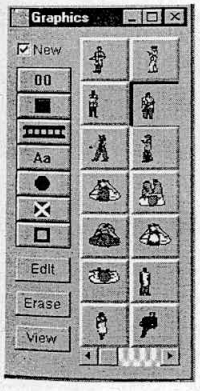


Figura 2.35 Entidad del modelo.

2. *BUILD (Locations)*

Definir las *Instalaciones* Zona_descanso y Servicio como Zonas. En la ventana *Graphics*, corresponden al rectángulo negro. Verificar que Cola_entrada y Cola_salida estén definidas como *Queue*.



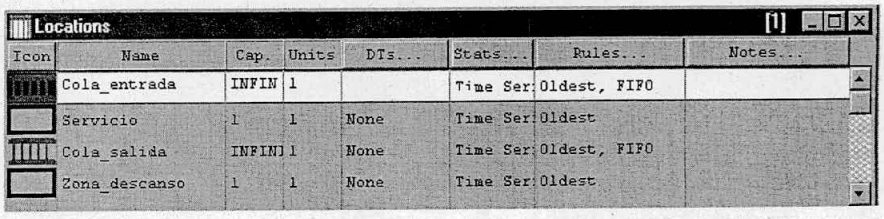


Figura 2.36 Instalaciones del modelo.

3. *BUILD (Arrivals)*

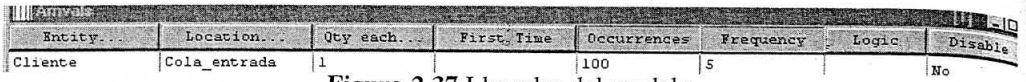


Figura 2.37 Llegadas del modelo.

4. *BUILD (Variables)*

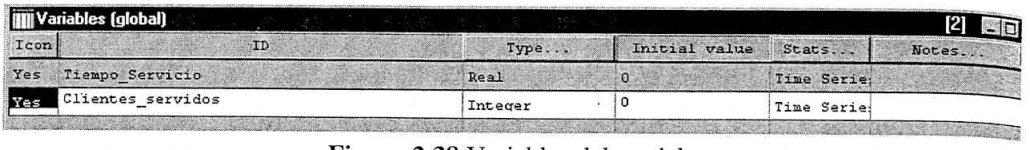


Figura 2.38 Variables del modelo.

5. *BUILD (Path Networks)*

Definir dos *Redes de recorrido* para los *Recursos*. La Red 1 define la trayectoria del primer recurso y la Red 2, la del segundo.

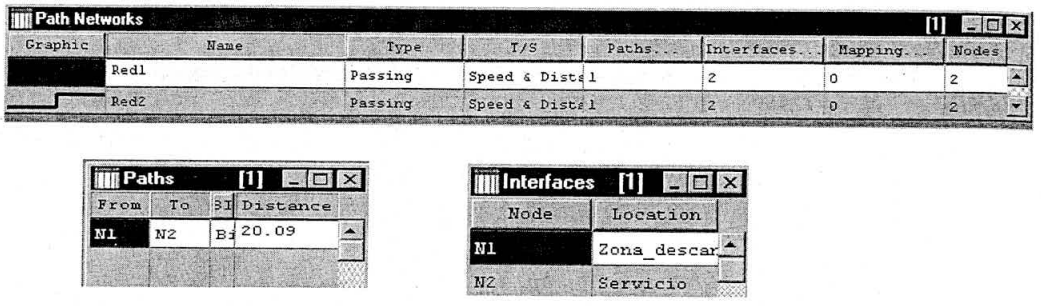


Figura 2.39 Redes del modelo.

6. *BUILD (Resources)*

Definir las *Especificaciones* de cada uno de los *Recursos*, dejando la Zona de descanso como Nodo *Hogar (Home)* y como lugar de descanso durante las pausas y cuando estén desocupados.

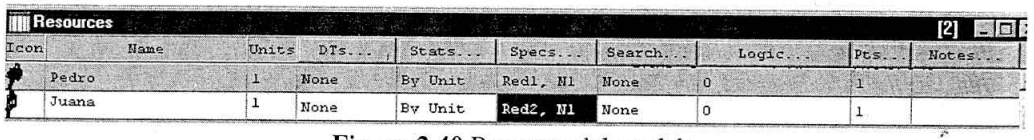


Figura 2.40 Recursos del modelo.

7. *BUILD (Shifts)*

Con este menú se pueden definir los *Turnos*. Al entrar aparecen dos opciones: *Define* y *Assign*. Escoger la primera para definir el horario del *Turno*.

Comenzar oprimiendo el botón *Options-Define Shift* en la ventana *Shift Editor*. Para programar *Turno* el lunes *(Monday)* de 8 a.m a 4 p.m., colocar el indicador del *mouse* sobre la línea de las 8 y arrastrar oprimiendo el botón izquierdo hasta las 4 p.m. Con las flechitas que aparecen en las ventanas de *Begin Time*, hacer clic sobre la hora y colocar el 8 con las flechas; luego, hacer clic sobre los minutos y con las flechas colocar 00. Repetir el procedimiento con *End Time* (4:00). Debe aparecer una franja azul desde las 8 hasta las 4. Para borrar, seleccionar con doble clic; debe aparecer un recuadro amarillo. Ir a *Edit* y oprimir *Delete*. Se debe tener cuidado con el ratón o *Mouse*, especialmente al definir las *Pausas (Breaks)*. Para definir una *Pausa,* en la ventana *Shift Editor,* ir a *Options* y oprimir *Define Break.* Programar dos *Pausas*: una de 10:00 a 10:20 a.m. y otra de 1:00 a 1:20 p.m. Con las flechitas de *Begin Time y End Time,* hacer coincidir el comienzo y el final de las *Pausas*. Deben aparecer dos franjas rojas. Ahora, hacer doble clic sobre la zona azul; debe aparecer un recuadro amarillo. Ir a *Edit*, oprimir el botón *Duplicate Monday* y señalar sobre el martes. Repetir la operación para los demás días de la semana. Después ir a *File,* oprimir *Save As* y guardar en el disco C:, Temp., con el nombre "Segundo Turno". Usted ya debe estar en capacidad de definir un "Tercer Turno" con horario de lunes a sábado de 4:00 p.m. a 11:59 p.m., y pausas de 7:00 a 7:20 p.m. y de 10:00 a 10:20 p.m.

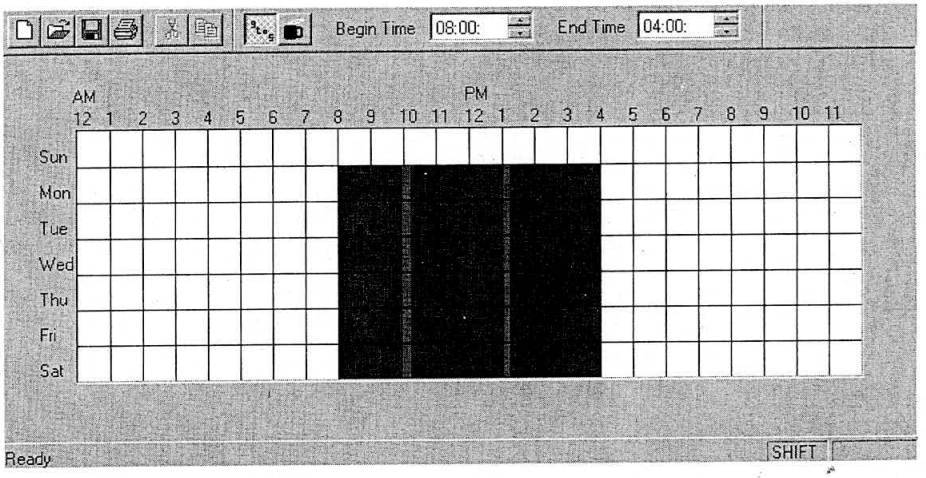


Figura 2.41 Turno de ocho horas con dos descansos de veinte minutos.

Para *Asignar* los turnos, ir a *Build-Shifts-Assign* y oprimir el botón *Shift Files*; luego, oprimir el botón *Add*. Se debe abrir la ventana *Select Shift File*, donde aparecen los turnos definidos previamente. Seleccionar el co-

rrespondiente, llenar el espacio que corresponde al minuto en que comienza el turno en la simulación y oprimir *Done*. Al oprimir el botón *Logic*, seleccionar *Break*. Se debe abrir la ventana *Break Logic*. Escribir *Display* "Hola, mi nombre es Juanita, trabajo en el tercer turno, vuelvo en 20 minutos".

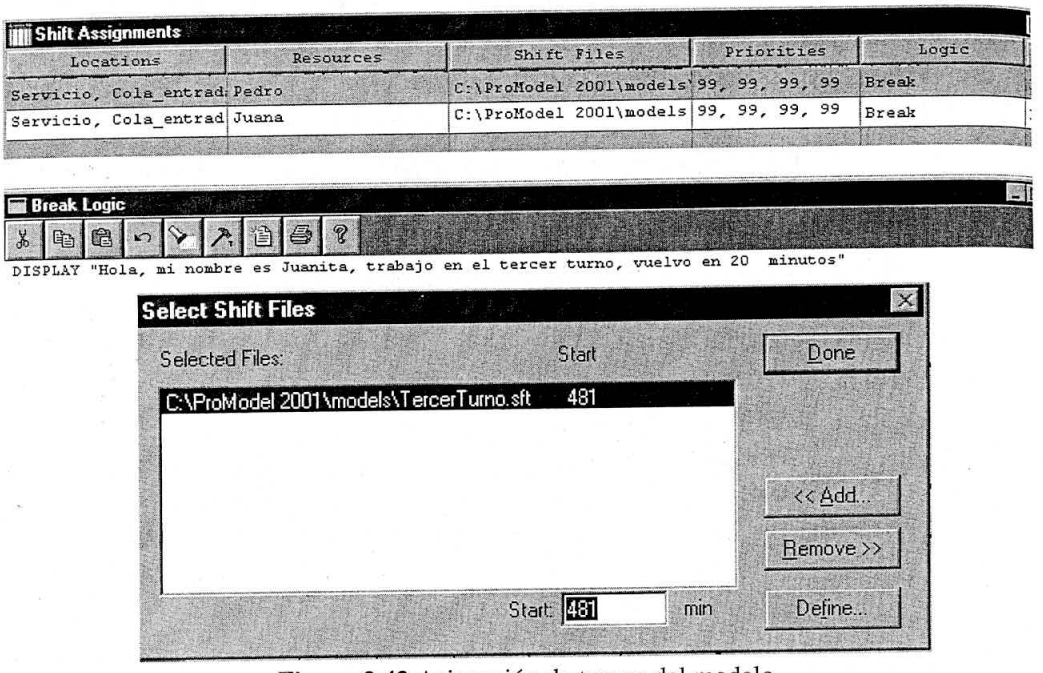


Figura 2.42 Asignación de turnos del modelo.

8. *BUILD (Processing)*

Entity	*Loc.*	*Operation*	*Output*	*Dest.*	*Rul*	*Logic*
Cliente	Cola_entrada	Tiempo_servicio = **CLOCK(MIN)**	Cliente	Servicio	First1	

Cliente	Servicio	**WHILE** Tiempo_servicio < 480 **DO**
		{ **GET** Pedro
		WAIT E(5)
		INC Clientes_servidos
		FREE Pedro}
		IF Tiempo_servicio = 480 **THEN**
		{ **DISPLAY "FIN SEGUNDO**
		DISPLAY "FIN SEGUNDO
		TURNO" }
		WHILE Tiempo_servicio > 480 **DO**
		{ **GET Juana**
		WAIT E(5)
		INC Clientes_servidos
		FREE Juana
		Tiempo_servicio = **CLOCK(MIN)**}
			Cliente	Cola_salida	First1	

Cliente	Cola_salida		Cliente	EXIT	First1	

9. *SIMULATION (Options)*

En la pantalla aparece el botón denominado *Browser*. Al oprimirlo, aparece la ventana *Examinar para encontrar carpeta*. Hacer doble clic sobre el ícono del disco (C:), buscar el archivo *Temp* y oprimir *Aceptar*. En *Output path* debe aparecer "C:\Temp". Verificar que en el punto *Define run lenght by* aparezca *Weekly Time*, que el tiempo esté en minutos y los reportes, en *estándar*. Correr el modelo durante 16 horas: *Sim Begin* 8:00, *Sim End* 11:59. Oprimir el botón *Run*.

El *Proceso* de este modelo ha sido programado para visualizar el trabajo de *Recursos* en los dos turnos. Al correrlo se observa una acumulación de clientes en el servicio, pero las estadísticas y los contadores indican los resultados. No se ve el desplazamiento de los clientes porque el programa ejecuta todas las instrucciones de *WHILE DO* en *Operation* antes de pasar al *Output*. Como ejercicio se sugiere que, en *Shift-Assign* para el recurso Juana, se coloque *Yes* en *Disable* para inhabilitar el tercer turno y en *Processing* GET Pedro; *WAIT* E(5); *INC* Clientes_servidos; *FREE* Pedro, con el fin de dejar solamente al *Recurso* Pedro en el segundo turno, como se ilustra más adelante.

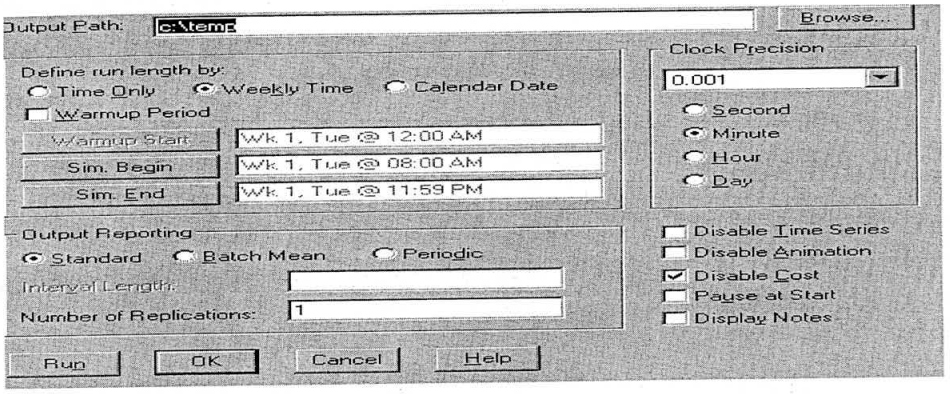


Figura 2.43 Opción para definir el tiempo de la simulación del modelo.

Entity	*Loc.*	*Operation*	*Output*	*Dest.*	*Rul*	*Logic*
Cliente	Cola_entrada		Cliente	Servicio	First1	
Cliente	Servicio	**GET** Pedro **WAIT** E(5) **INC** Clientes_servidos **FREE** Pedro	Cliente	Cola_salida	First1	
Cliente	Cola_salida		Cliente	EXIT	First1	

Modelo 9. Mediciones de tiempos y de inventarios en proceso

En este modelo se pretende mostrar cómo se puede calcular el tiempo de ciclo de un producto, entendido éste como el tiempo que transcurre desde que entra la materia prima hasta que sale como producto terminado. El tiempo de ciclo tiene dos componentes principales: el tiempo de operación y el tiempo de espera. Además se quiere mostrar cómo llevar la cuenta de los inventarios en proceso (WIP) en cada una de las operaciones y en el sistema total.

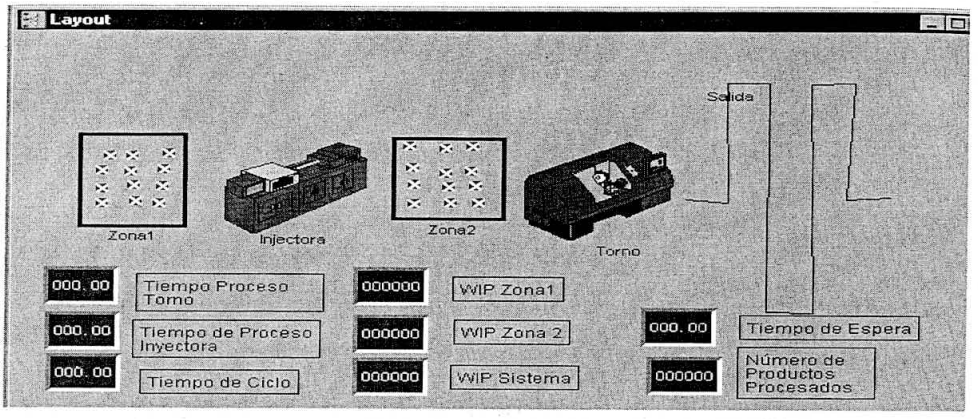
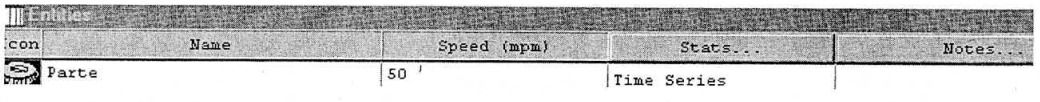


Figura 2.44 Layout del modelo en donde se destacan dos zonas de almacenamiento.

Comenzar, como siempre, haciendo clic en *File-New-Title*. Luego, escribir "Modelo-Tiempos-su nombre" en el renglón correspondiente al título. Verificar que la *librería* sea INDUSTRY.GLB, definir *Units* en minutos y *Distance* en metros. Salvar mediante *File-Save AS* en el disco (C:), en *Temp,* archivo "Tiempos-su nombre" y hacer clic en *Save*. No olvidar salvar cada cinco minutos.

1. *BUILD (Entities)*



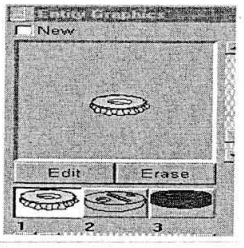
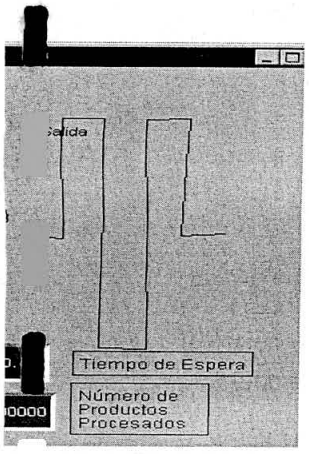


Figura 2.45 Entidad del modelo con tres gráficos.

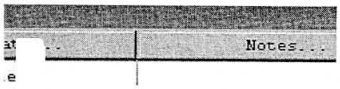
ri en proceso

: calcular el tiempo de
o ıe transcurre desde
rc..ucto terminado. El
el tiempo de operación
ın llevar la cuenta de
as operaciones y en el



ɔr de almacenamiento.

N *w-Title*. Luego, es-
o espondiente al títu-
lefinir *Units* en minu-
A en el disco (C:), en
ve. No olvidar sal-



on tres gráficos.

2. *BUILD (Locations)*

El proceso comienza en la *Zona 1* de almacenamiento de materiales para la *Inyectora*, con una capacidad de12 partes. En la *Zona 2*, se almacenan las partes semiprocesadas antes de pasar al *Torno*. También tiene una capacidad de 12. Al final hay una *cola*, llamada *Salida*, que permite visualizar los productos terminados.

Locations [3]

Icon	Name	Cap.	Units	DTs...	Stats...	Rules...	Notes...
	Injectora	1	1	None	Time Serie	Oldest	
	Torno	1	1	None	Time Serie	Oldest	
	Zonal	12	1		Time Serie	Oldest	
	Zona2	12	1	None	Time Serie	Oldest	
	Salida	INFINI	1	None	Time Serie	Oldest, FIFO	

Figura 2.46 Instalaciones del modelo.

3. *BUILD (Arrivals)*

Arrivals [1]

Entity...	Location...	Qty each...	First Time	Occurrences	Frequency	Logic	Disable
Parte	Zonal	1		500	p(3)		No

Figura 2.47 Llegadas del modelo.

Las *partes* llegan a la *Zonal* con una frecuencia definida por la distribución de Poisson con media 3. La distribución de Poisson es especial para simular llegadas aleatorias discretas.

4. *BUILD (Attributes)*

Attributes [1]

ID	Type...	Classification...	Notes...
TEntradaSis	Real	Ent	
TSalidaSis	Real	Ent	

Figura 2.48 Atributos del modelo.

En estos dos *Atributos*, se guarda el tiempo de entrada y el tiempo de Salida de la *Entidad parte* al sistema.

5. *BUILD (Variables)*

Variables (global)

Icon	ID	Type...	Initial value	Stats...	Note
	TCiclo_producto	Real	0	Time Serie	
Yes	WIPZona1	Integer	0	Time Serie	
Yes	WIPZona2	Integer	0	Time Serie	
Yes	WIPSistema	Integer	0	Time Serie	
Yes	PProcesadas	Integer	0	Time Serie	
Yes	TProcTORNO	Real	e(10)	Time Serie	
Yes	TProcINY	Real	e(3)	Time Serie	
Yes	TEspera	Real	0	Time Serie	

Figura 2.49 Variables del modelo.

Los tiempos de proceso de la *Inyectora* y del *Torno* se van a guardar en las variables reales, *TProcTORNO* y *TProcINY*, definidos con distribución exponencial, con media 10 y 3 respectivamente. La distribución exponencial es especial para definir tiempos de servicio.

6. *BUILD (Processing)*

El *Tiempo de ciclo* es el tiempo que la *parte* ha permanecido en el sistema. Si a ese tiempo se le resta el *Tiempo de proceso* de las dos máquinas, resulta el *Tiempo de espera*.

Entity	*Location*	*Operation*	*Output*	*Destination*	*Rule*	*Move Logic*
Parte	Zona1	TEntradaSis= **CLOCK(MIN)** **INC** WIPZona1 **INC** WIPSistema	Parte	Inyectora	First1	Move For 1
Parte	Inyectora	**WAIT** TProcINY **DEC** WIPZona1 **GRAPHIC** 2	Parte	Zona2	First1	Move For 1
Parte	Zona2	**INC** WIPZona2	Parte	Torno	First1	Move For 1
Parte	Torno	**WAIT** TProcTORNO **DEC** WIPZona2 **GRAPHIC** 3	Parte	Salida	First1	Move For 1
Parte	Salida	TsalidaSis = **CLOCK(MIN)** TCiclo_Producto=TsalidaSis-TentradaSis Tespera=Tciclo_Producto – (TProcINY+TProcTORNO) **DEC** WIPSistema **INC** PProcesadas	Parte	EXIT	First1	Move For 1

7. *SIMULATION (Options)*

En la pantalla aparece un botón llamado *Browser*. Al oprimirlo aparece la ventana *Examinar para encontrar carpeta*. Hacer doble clic sobre el ícono del disco (C:), buscar el archivo *Temp* y oprimir *Aceptar*. En *Output path* debe aparecer "C:\Temp". Verificar que en el punto *Define run lenght by* aparezca *Time Only*, que el tiempo esté en minutos y los reportes, en *estándar*. Oprimir el botón *Run*.

Cuando la simulación empieza a correr, aparece el letrero *Information* en la parte superior de la ventana. Al hacer clic en este botón, se abre una ventana. Al final se encuentra un botón llamado *Dynamic Plots*. Al señalarlo aparece *New, Configurations*. Seleccione *New*. Se abre una ventana. Seleccione el botón *Stats to Plot*. Buscar el ícono de archivo *Variables* y oprimirlo. En la lista que aparece debajo de la variable *TCiclo_Producto*, seleccionar el cuadrado situado en frente de *Current Value*. Hacer lo mismo para la variable *WIPSistema*. A medida que la simulación transcurre, se puede observar el comportamiento de estas dos variables.

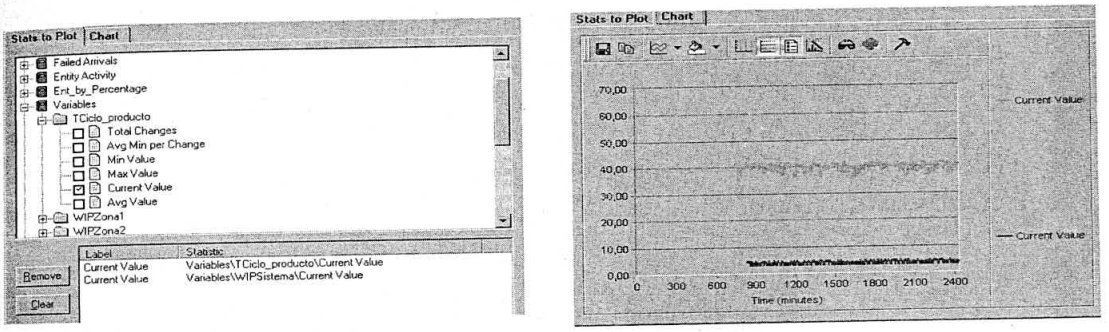


Figura 2.50 Gráficos dinámicos de la simulación.

Modelo 10. Uso de herramientas Stat::Fit y Graphics Editor

En este modelo se pretende ilustrar el uso de la herramienta *Stat::Fit*, de gran ayuda cuando se tienen datos reales (por ejemplo, tiempos de llegada a una cola, tiempos de espera, etc.) y se quieren conocer los parámetros de las distribuciones de densidad-probabilidad que más se ajustan a dichos datos. Además, se quiere mostrar el uso del *Editor Gráfico* que ayuda a definir *Librerías* más completas o más al gusto del usuario.

Suponga que se han tomado los datos de la hora de llegada de clientes a un banco en un periodo de una hora (9:00 a las 10:00). Estos datos se ilustran en la tabla 2.3. Los cuatro primeros clientes llegaron a las 9:00, el quinto y el sexto a las 9:01, etc.

Número	Hora	Número	Hora	Número	Hora	Número	Hora
1	9,00	11	9,04	21	9,22	31	9,36
2	9,00	12	9,05	22	9,22	32	9,40
3	9,00	13	9,06	23	9,22	33	9,45
4	9,00	14	9,07	24	9,24	34	9,45
5	9,01	15	9,07	25	9,25	35	9,46
6	9,01	16	9,10	26	9,27	36	9,47
7	9,02	17	9,11	27	9,30	37	9,48
8	9,02	18	9,15	28	9,32	38	9,50
9	9,03	19	9,17	29	9,33	39	9,52
10	9,03	20	9,20	30	9,34	40	9,55
						41	9,57
						42	9,58
						43	9,59

Tabla 2.3 Llegada de clientes a un banco, en la primera hora de apertura.

Comenzar, como siempre, haciendo clic en *File-New-Title*. Luego, escribir "Modelo-Herramientas-su nombre" en el renglón correspondiente al título. Verificar que la *librería* sea *PROMODEL.GLB.*, definir *Units* en minutos y *Distance* en metros. Salvar mediante *File-Save AS* en el disco (C:), en *Temp,* archivo "Herramientas-su nombre" y hacer clic en *Save*. No olvidar salvar cada cinco minutos.

1. *TOOLS-Stat::Fit*

Al oprimir el botón *Stat::Fit*, aparece una ventana con este nombre. Oprimir el botón *File*. Se abre una ventana más pequeña *Document1-Input*, donde va a introducir los datos de entrada de clientes al banco, que aparecen en la tabla. Al terminar, ir a *File-Save Input* y salvar en C: *Output-llegadas.dat*. Oprimir el botón *Auto Fit*. Debe aparecer una ventana *Auto Fit*. Seleccionar el chulo verde *OK*. Debe abrirse otra ventana en donde se listan las distribuciones en orden de ajuste de los datos. Seleccionar la distribución que según sus conocimientos se ajuste mejor. Si no conoce su proceso, simplemente seleccione la primera. Para este ejemplo, seleccionar la *Exponencial*. Al hacer clic, se abren dos ventanas *Comparison Graph*, donde se muestran los datos y la curva de la función y *Difference Graph* donde se muestran los residuos acumulados. En este caso los parámetros de llegadas se ajustan a una distribución *Exponencial* (9, 0.238).

	Número	Hora
22	31	9,36
9,22	32	9,40
2	33	9,45
4	34	9,45
9,25	35	9,46
7	36	9,47
0	37	9,48
9,32	38	9,50
3	39	9,52
	40	9,55
	41	9,57
	42	9,58
	43	9,59

ra hora de apertura.

New-Title. Luego, es-
ó correspondiente al
LD., definir *Units* en
Save AS en el disco
a r clic en *Save*. No

e nombre. Opri-
ocument1-Input,
l banco, que apare-
r en C: *Output-*
na ventana *Auto-*
entana en donde se
eccionar la dis-
Si no conoce su
ejemplo, seleccio-
o varison Graph,
ence Graph donde
rámetros de llega-

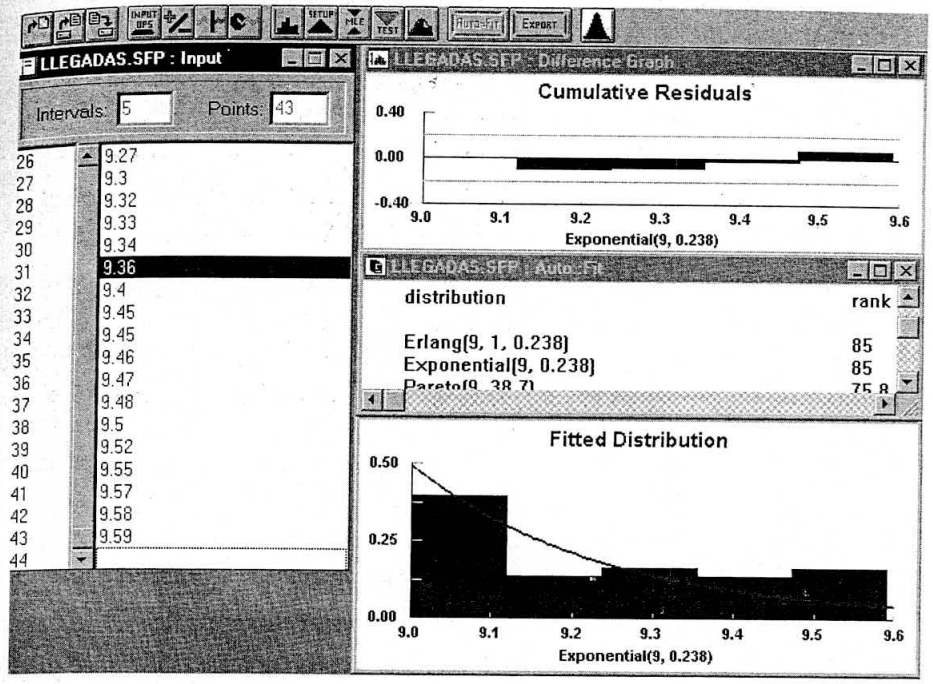


Figura 2.51 Curva de ajuste de datos mediante *Stat::Fit*.

2. *TOOLS (Graphic Editor)*

Ahora, vamos a fusionar las *Librerías PROMODEL.GLB y PERSONAS.GLB* en una nueva que se llamará *PRODUCCION.GLB*. Para realizarlo, ir al menú *Tools-Graphic Editor*.A

Al hacer clic se abre la ventana del *Editor*. Oprimir el botón *File* del *Editor* y guardar mediante *Save As* con el nombre *PRODUCCION*. Debe aparecer una ventana como la que se muestra a continuación, que contiene todos los íconos de la *Librería PROMODEL 2001*. Ahora se agregarán los íconos de la *Librería PEOPLE*.

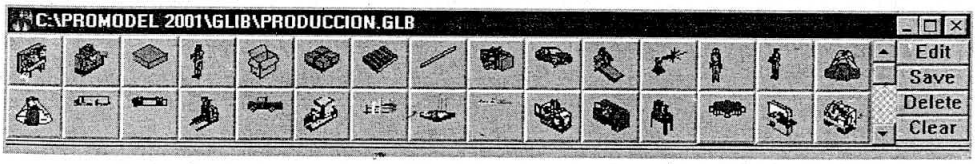


Figura 2.52 Librería resultante de la fusión de dos existentes.

Para abrir la *Librería PEOPLE*, ir a *File-Open* en el *Editor Gráfico* y seleccionar *PEOPLE.GLB*. Sobre la ventana del *Editor Gráfico,* abrir simultáneamente las *Librerías PEOPLE* y *PRODUCCION*, dejando la primera a la izquierda y la otra a la derecha.

Para trasladar el contenido de la *Librería PEOPLE* a la que se creó y que se llamó *PRODUCCION*, seleccionar el primer ícono de la *Librería PEOPLE* y, con el botón izquierdo del *mouse* oprimido, arrastrarlo hasta la ventana de la *Librería PRODUCCION*. Si el tamaño de la imagen es muy grande, adecuarlo con el *mouse*, arrastrando los cuadraditos que aparecen en los vértices del rectángulo que rodea la figura. A la derecha, en la parte superior de la ventana de la *Librería PRODUCCION*, junto a los letreros *Edit, Save,Delete, Clear,* aparece un botón cuadrado entre dos flechas. Oprimirlo hacia abajo y buscar el final. Allí debe aparecer un cuadrado sin imagen. Ahora, oprimir *Save* (el de la ventana de la *Librería PRODUCCION*) y el ícono que está señalado debe aparecer en el cuadrado vacío. Oprimir *Clear* para borrar el ícono que está en la ventana.

Ir a la ventana de la *Librería PEOPLE*, seleccionar y arrastrar el segundo ícono, oprimir *Save* y luego *Clear*. Repetir la operación anterior hasta que se trasladen todas las imágenes de la *Librería PEOPLE* a la *Librería PRODUCCION*.

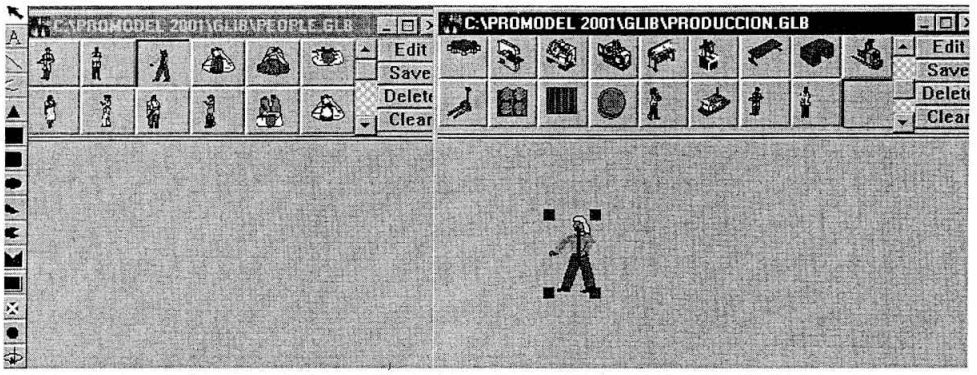


Figura 2.53 Forma como se deben colocar las librerías; la nueva debe ir a la derecha.

Para dibujar un ícono propio, en este caso una carita amarilla, utilizar la barra de herramientas de dibujo que aparece a la izquierda del *Editor*. Dibujar la carita en la ventana de la *Librería PRODUCCION*, oprimir *Save* y luego *Clear*.

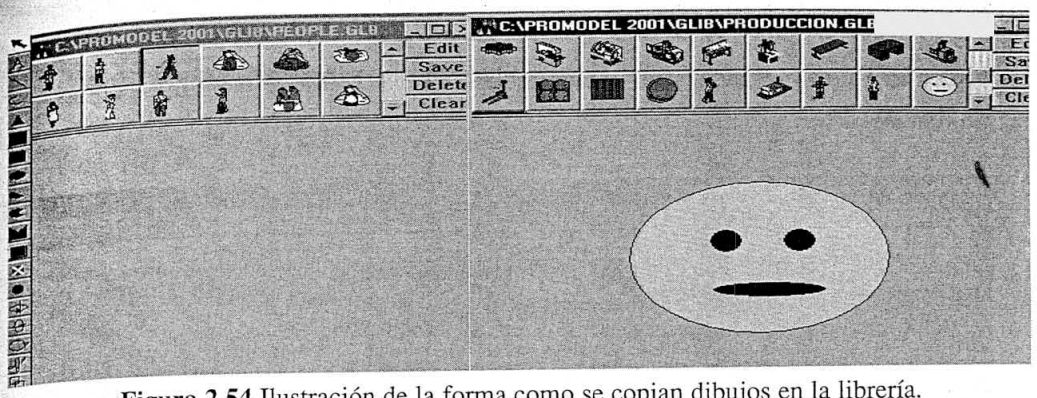


Figura 2.54 Ilustración de la forma como se copian dibujos en la librería.

También puede traer imágenes de otros programas. Por ejemplo, desde *Power Point- Insertar-Insertar Imágenes- Insertar Imágenes Prediseñadas-Animales*, se puede *seleccionar* un asno y *copiarlo*. Ir luego al *Editor Gráfico,* oprimir *Edit* y *Copy to Clipboard*. En la ventana de la *Librería PRODUCCION* debe aparecer el asno. Oprimir *Save y Clear*. Con este procedimiento se pueden construir *librerías* muy completas. No obstante, se debe recordar que, como se trata de imágenes, éstas ocupan mucha memoria y se dificulta su grabación. Para guardar la *Librería PRODUCCION.GLB*, ir a *File* del *Editor Gráfico* y oprimir *Save*. Cerrar el *Editor Gráfico*.

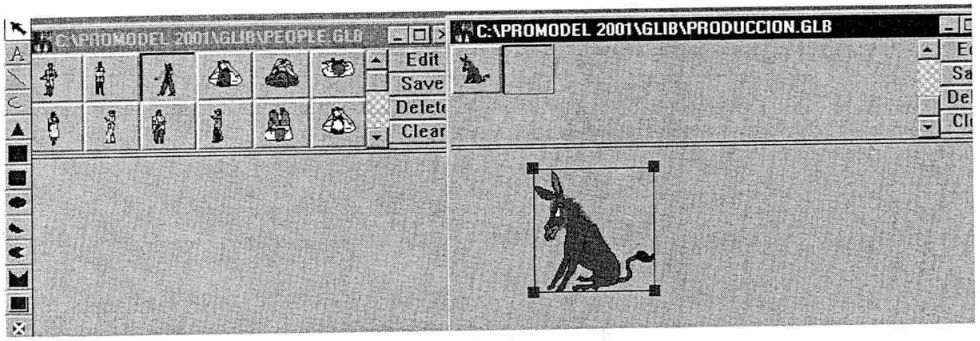


Figura 2.55 Importación de imágenes predeterminadas a la librería.

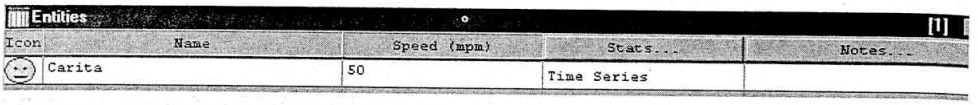
Aprendidas las dos herramientas *Stat::Fit* y *Graphic Editor*, seguir el procedimiento tradicional para construir modelos.

3. *BUILD (General Information)*

Oprimir el botón *Graphic Library File*. Se abre una ventana con todas las librerías disponibles. Seleccionar *PRODUCCION.GLB* y dar *OK*.

4. *BUILD (Entities)*

Con el fin de verificar que la carita quedó en la *Librería*, se coloca como entidad.



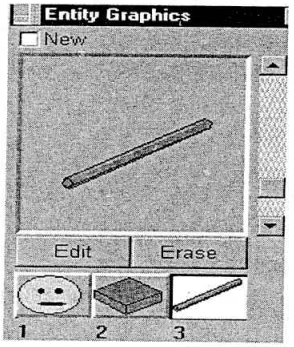


Figura 2.56 Entidad del modelo con tres gráficos.

Al quitar el chulo en *New*, se definen dos gráficas más, que aparecerán durante la simulación cuando se le solicite mediante la instrucción *Graphic 2 o Graphic 3*.

5. *BUILD (Locations)*

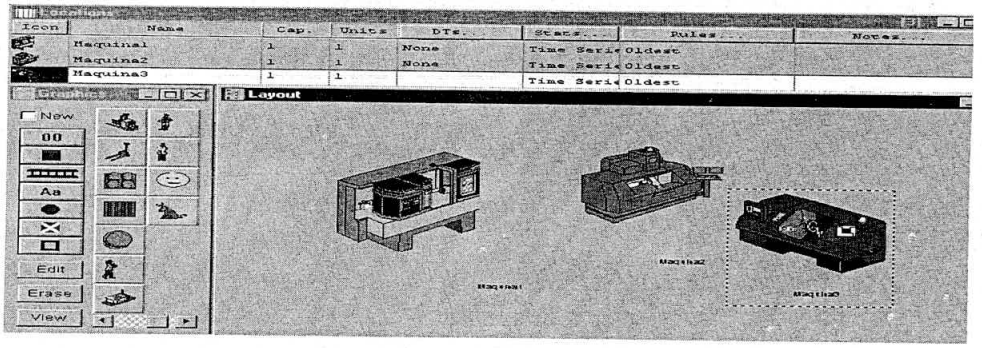


Figura 2.57 Instalaciones del modelo.

6. *BUILD (Arrivals)*

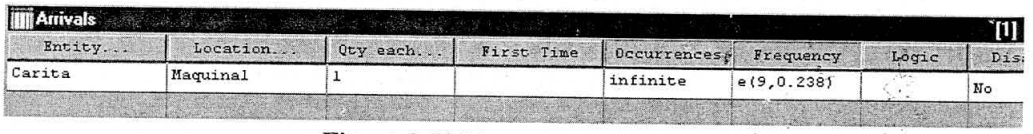


Figura 2.58 Llegadas del modelo.

Aquí se ha colocado la distribución y los parámetros que más se ajustan a los datos de llegada.

Las llegadas al sistema se hacen siguiendo una distribución *Exponencial* (9, 0.238).

7. *BUILD (Variables)*

Variables (global)

Icon	ID	Type...	Initial value	Stats...	Notes
Yes	t1	Real	E(5,1)	Time Serie:	
Yes	t2	Real	E(20,2)	Time Serie:	
Yes	t3	Real	E(2,0.5)	Time Serie:	
Yes	r1	Real	0	Time Serie:	
Yes	r2	Real	0	Time Serie:	
Yes	r3	Real	0	Time Serie:	
Yes	TH	Real	0	Time Serie:	

Figura 2.59 Variables del modelo.

En las variables *t1, t2, t3* se guardan los tiempos de operación de las máquinas 1, 2 y 3, respectivamente. En las variables *r1, r2, r3* se guardan las tasas de producción de las máquinas 1, 2 y 3. En la variable *TH* se guarda el *Throughput* del sistema.

8. *BUILD (Processing)*

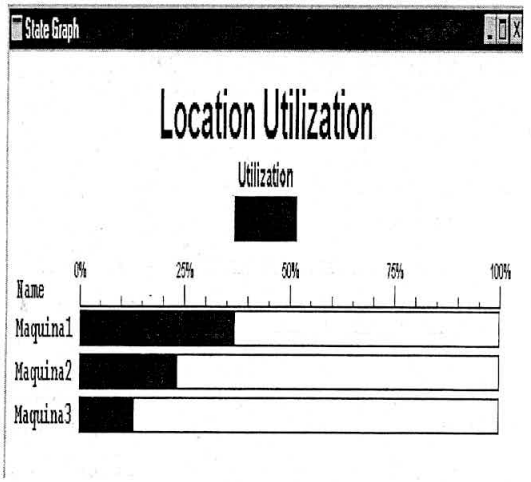
Entity	*Location*	*Operation*	*Output*	*Destination*	*Rule*	*Move Logic*
Carita	Máquina1	r1 = 1 / t1 **WAIT** t1 **GRAPHIC** 2	Carita	Máquina2	First1	Move For 1
Carita	Máquina2	r2 = 1 / t2 **WAIT** t2 **GRAPHIC** 3	Carita	Máquina3	First1	Move For 1

Carita	Máquina3	r3 = 1 / t3
		WAIT t3
		IF r1 < r2 **AND** r1 < r3 **THEN**
		{
		TH=r1
		}
		ELSE
		IF r2 < r3 **THEN**
		{
		TH=r2
		}
		ELSE
		TH=r3
			Carita	EXIT	First1	Move For 1

9. *SIMULATION (Options)*

En la pantalla aparece un botón denominado *Browser*. Al oprimirlo aparece la ventana *Examinar para encontrar carpeta;* hacer doble clic sobre el ícono del disco (C:). Buscar el archivo *Temp* y oprimir *Aceptar*. En *Output path* debe aparecer "C:\Temp". Verificar que en el punto *Define run lenght by* aparezca *Time Only,* que el tiempo esté en minutos y los reportes, en *estándar*. *Run Hours* 8. Oprimir el botón *Run*.

Cuando aparezca el letrero *Simulation Complete Do you want to se the results?*, responder *Sí*. Oprimir el botón de barras horizontales, debajo de *Help*. Se abre la ventana *State and loçalization Graph* en donde aparece señalado en azul *Location Utilization*. Al darle *OK* se abre la ventana *State Graph* donde se muestran los porcentajes de utilización de las máquinas. El cuello de botella corresponde a la más utilizada. En este caso la máquina 1, que tiene la menor tasa de producción.



General Report

Entity Name	In Move Logic	Wait For Res, etc.	% In Operation	% Blocked
Carita	9.83	0.00	90.17	0.00

VARIABLES

Variable Name	Total Changes	Average Minutes Per Change	Minimum Value	Maximum Value	Current Value	Average Value
t1	0	0.00	4.57	4.57	4.57	4.57
t2	0	0.00	2.93	2.93	2.93	2.93
t3	0	0.00	1.67	1.67	1.67	1.67
r1	39	12.12	0	0.21	0.21	0.21
r2	39	12.25	0	0.34	0.34	0.33
r3	38	12.54	0	0.59	0.59	0.58
TH	38	12.58	0	0.21	0.21	0.21

Figura 2.60 Resultados de la simulación del modelo.

Modelo 11. Fábrica de monedas de centavo (Penny Two)

Este modelo ilustra un ejemplo que trae el libro *Factory Physics*, de Wallace Hoppp y Mark Spearman, de una fábrica donde se hacen monedas de centavo estadounidenses (*pennies*). El proceso productivo cuenta con cuatro estaciones de trabajo. En la primera estación hay una troqueladora que corta los círculos; su tiempo de operación es 2 horas. En la segunda estación hay dos troqueladoras que estampan la cara y el sello de la moneda. El tiempo de operación de cada una es 5 horas. En la tercera estación hay seis

troqueladoras que hacen los anillos de los bordes de las monedas; el tiempo de operación de cada una es 10 horas. La cuarta y última estación tiene dos pulidoras que retiran las rebabas y brillan las monedas; el tiempo de operación de cada una es 3 horas. El flujo de los materiales sigue el mismo orden de las estaciones; el trabajo se realiza en la máquina que se encuentre disponible.

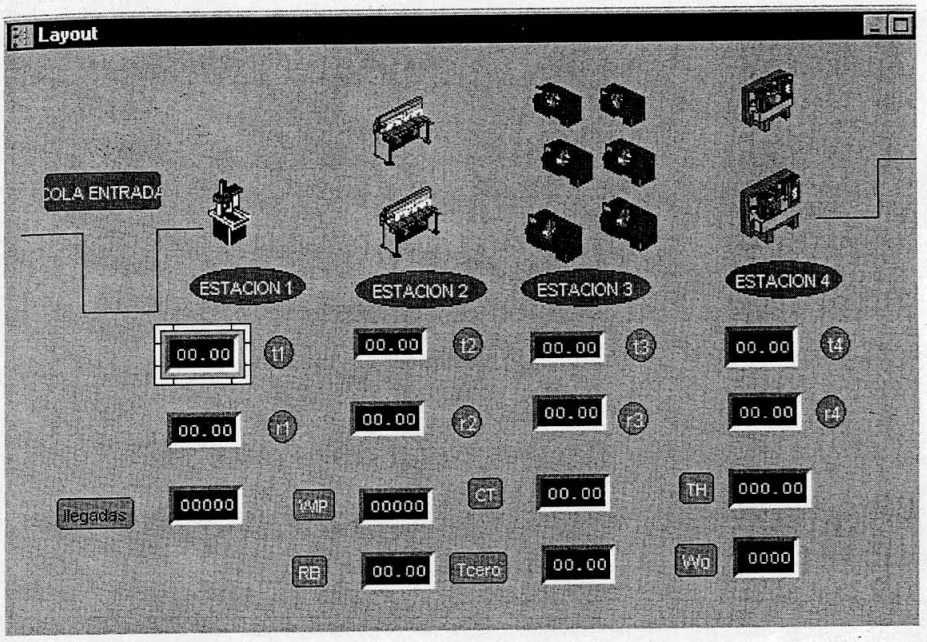


Figura 2.61 Layout del modelo con cuatro estaciones de trabajo.

Comenzar, como siempre, haciendo clic en *File-New-Title*. Luego, escribir "Modelo-Penny2-su nombre", en el renglón correspondiente al título. Verificar que la *librería* sea *PROMODEL.GLB*, definir *Units* en horas y *Distance* en metros. Salvar mediante *File-Save AS* en el disco (C:), en *Temp*, archivo "Penny2-su nombre" y hacer clic en *Save*. No olvidar salvar cada cinco minutos.

1. *BUILD (Entities)*

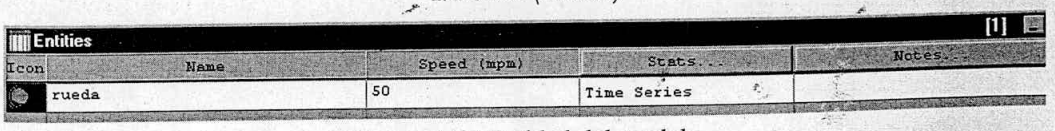


Figura 2.62 Entidad del modelo.

2. *BUILD (Locations)*

Locations [1]

Icon	Name	Cap.	Units	DTs...	Stats...	Rules...	Notes...
	estacion_1	1	1		Time Serie	Oldest	
	estacion_2	1	2	None	Time Serie	Oldest, First	
	estacion_2.1	1	1	None	Time Serie	Oldest	
	estacion_2.2	1	1	None	Time Serie	Oldest	
	estacion_3	1	6	None	Time Serie	Oldest, First	
	estacion_3.1	1	1	None	Time Serie	Oldest	
	estacion_3.2	1	1	None	Time Serie	Oldest	
	estacion_3.3	1	1	None	Time Serie	Oldest	
	estacion_3.4	1	1	None	Time Serie	Oldest	
	estacion_3.5	1	1	None	Time Serie	Oldest	
	estacion_3.6	1	1	None	Time Serie	Oldest	
	estacion_4	1	2	None	Time Serie	Oldest, First	
	estacion_4.1	1	1	None	Time Serie	Oldest	
	estacion_4.2	1	1	None	Time Serie	Oldest	
	cola_entrada	INFINI	1	None	Time Serie	Oldest, FIFO	
	cola_salida	INFINI	1	None	Time Serie	Oldest, FIFO	

Figura 2.63 Instalaciones del modelo.

3. *BUILD (Macros)*

En este modelo se va a controlar el número de llegadas mediante una *Macro* llamada *arrivos*.

Macros [1]

ID	Text...	Option
arrivos	llegadas□□	RTI

RTI Definition for arrivos

Parameter Name: arrivos

Prompt:

○ Unrestricted Text

○ Record Range

◉ Numeric Range

From 1 To 20

OK Cancel Help

Figura 2.64 Ejemplo de definición de un macro.

4. *BUILD (Arrivals)*

Arrivals [1]

Entity...	Location...	Qty each...	First Time	Occurrences	Frequency	Logic	Dis
rueda	cola_entrada	arrivos	0	1	1		No

Figura 2.65 Llegadas del modelo.

5. *BUILD (Variables)*

El nombre de las variables corresponden a conceptos de Física de Planta:

To = t1+t2+t3+t4 = Tiempo global del sistema

TH = min{r1,r2,r3,r4} = *Throughput* = Tasa de salida del sistema

WIP = *Work In Process* = Número de unidades de materia prima y productos en proceso

CT = Tiempo de ciclo = Tiempo que gasta la materia prima desde que entra hasta que sale como producto terminado.

Ley de Little = TH = (WIP)/CT

Variables (global)

Icon	ID	Type...	Initial value	Stats...	Not
Yes	t1	Real	2	Time Serie	
Yes	t2	Real	5	Time Serie	
Yes	t3	Real	10	Time Serie	
Yes	t4	Real	3	Time Serie	
Yes	r1	Real	0	Time Serie	
Yes	r2	Real	0	Time Serie	
Yes	r3	Real	0	Time Serie	
Yes	r4	Real	0	Time Serie	
Yes	llegadas	Integer	1	Time Serie	
Yes	WIP	Integer	0	Time Serie	
No	tiempo_entrada	Real	0	Time Serie	
No	tiempo_salida	Real	0	Time Serie	
Yes	CT	Real	0	Time Serie	
Yes	TH	Real	0	Time Serie	
Yes	tcero	Real	0	Time Serie	
Yes	RB	Real	0	Time Serie	
Yes	Wo	Integer	0	Time Serie	

Figura 2.66 Variables del modelo.

6. *BUILD (Processing)*

Entity	*Location*	*Operation*	*Output*	*Destination*	*Rule*	*Mv*
rueda	Cola_entrada	tcero = t1 + t2 + t3 + t4 tiempo_entrada = **CLOCK**(HR) **INC** WIP	rueda	Estación1	First1	
rueda	Estación1	**WAIT** t1 r1 = 1/t1	rueda	Estación2	First1	
rueda	Estación2	**WAIT** t2 r2 = 2*1/t2	rueda	Estación3	First1	
rueda	Estación3	**WAIT** t3 r3 = 6*1/t3	rueda	Estación4	First1	

rueda	Estación4	**WAIT** t4
		r4 = 2*1/t4
		IF r1 < r2 **AND** r1 < r3 **AND** r1 < r4 **THEN**
		{RB = r1 }
		ELSE
		IF r2 < r3 **AND** r2 < r4 **THEN** {RB = r2}
		ELSE
		IF r3 < r4 **THEN** {RB = r3}
		ELSE
		RB = r4
			rueda	Cola_salida	First1	

rueda	Cola_salida	tiempo_salida = **CLOCK**(HR) CT = tiempo_salida - tiempo_entrada TH = WIP / CT Wo = tcero * RB	rueda	EXIT	First1	

7. *SIMULATION (Options)*

En la pantalla aparece un botón denominado *Browser*. Al oprimirlo aparece la ventana *Examinar para encontrar carpeta*. Hacer doble clic sobre el ícono del disco (C:), buscar el archivo *Temp* y oprimir *Aceptar*. En *Output path* debe aparecer "C:\Temp". Verificar que en el punto *Define run lenght by* aparezca *Time Only,* que el tiempo esté en minutos y los reportes, en *estándar*.

Ahora se quieren definir *Escenarios* de la cantidad de llegadas. Los *Escenarios* responden a la pregunta ¿qué pasa con las variables, si las entradas son...?

	Output	Destination	Rule	My
	ueda	Estación1	First1	
	ueda	Estación2	First1	
	rueda	Estación3	First1	
	ueda	Estación4	First1	
	ieda	Cola_salida	First1	
	eda	EXIT	First1	

r. Al oprimirlo apare-
er doble clic sobre el
r *ceptar*. En *Output*
no *Define run lenght*
tos y los reportes, en

lad de llegadas. Los
s ariables, si las en-

8. SIMULATION (Scenarios)

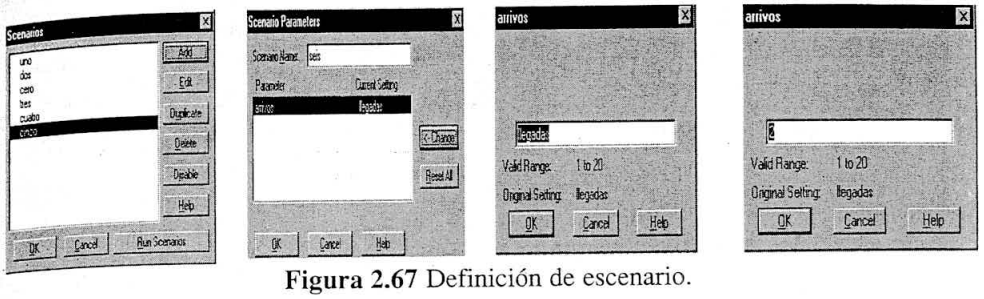


Figura 2.67 Definición de escenario.

Para definir *Escenarios*, proceder de la siguiente manera: en la ventana *Scenarios*, oprimir el botón *Add*. Se abre la ventana *Scenario Parameters*. En la ventanita superior, colocar el nombre que se va a dar a este *Escenario*; en este caso, *dos*. Oprimir el botón *Change*. Se abre la ventana *Arrivos*. En la banda central colocar 2, que corresponde a dos llegadas. Oprimir *OK*. En la ventana *Scenarios* aparece el nombre *dos*. Definir *Escenarios* para 3, 4, 10, 15 y 20 llegadas. Oprimir el botón *Run Scenarios*. Aceptar abrir las *Estadísticas*. Observar los valores de las variables en cada uno de los *Escenarios*.

Modelo 12. Mezcla de modelos

En este modelo se pretende mostrar cómo unir dos modelos mediante la instrucción *Merge*. Para poder unir un modelo A con un modelo B, la salida del primero debe coincidir con la entrada del segundo. Las *Entidades* que salen del modelo A y que entran en el modelo B deben ser las mismas. Los modelos que se presentan aquí tienen la mínima complejidad, debido a que sólo interesa el proceso de mezclado.

Modelo A

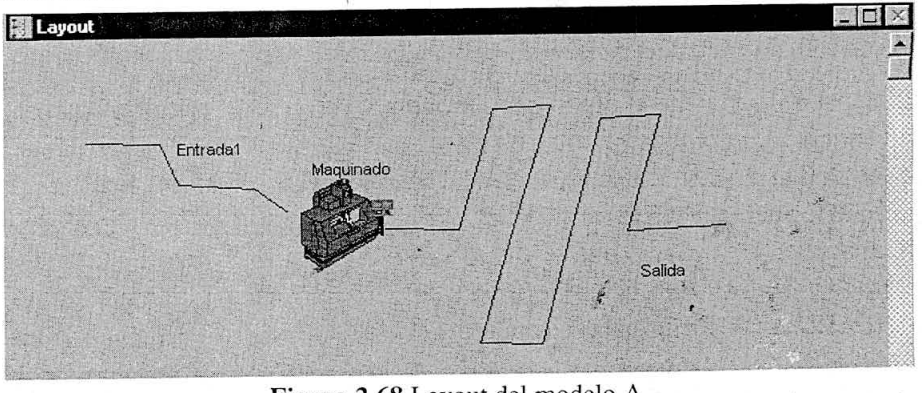


Figura 2.68 Layout del modelo A.

1. *BUILD (Locations)*

Locations [1]

Icon	Name	Cap.	Units	DTs...	Stats...	Rules...	Notes...
	Entrada1	INFINI	1	None	Time Serie	Oldest, FIFO	
	Maquinado	1	1	None	Time Serie	Oldest	
	Salida	INFINI	1	None	Time Serie	Oldest, FIFO	

Figura 2.69 Instalaciones del modelo A.

2. *BUILD (Entities)*

Definir una Gráfica 2.

Entities [1]

Icon	Name	Speed (mpm)	Stats...	Notes...
	Subprod	50	Time Series	

Figura 2.70 Entidad del modelo A.

3. *BUILD (Arrivals)*

Arrivals [1]

Entity...	Location...	Qty each...	First Time	Occurrences	Frequency	Logic	Disabl
Subprod	Entrada1	1	0	100	1		No

Figura 2.71 Llegadas del modelo A.

4. *BUILD (Processing)*

Entity	*Location*	*Operation*	*Output*	*Destination*	*Rule*	*My*
Subprod	Entrada1		Subprod	Maquinado		
Subprod	Maquinado	**WAIT** 5 **GRAPHIC** 2	Subprod	Salida		
Subprod	Salida		Subprod	EXIT		

5. *SIMULATION (Options)*

En la pantalla aparece el botón denominado *Browser*. Al oprimirlo aparece la ventana *Examinar para encontrar carpeta*. Hacer un doble clic sobre el ícono del disco (C:), buscar el archivo *Temp* y oprimir *Aceptar*. En *Output path* debe aparecer "C:\Temp". Verificar que en el punto *Define run lenght by* aparezca *Time Only,* que el tiempo esté en minutos y los reportes, en *estándar*. Oprimir *Run*. El modelo debe comenzar a correr. Guardarlo como *MODELO A* y abrir un nuevo modelo.

Modelo B

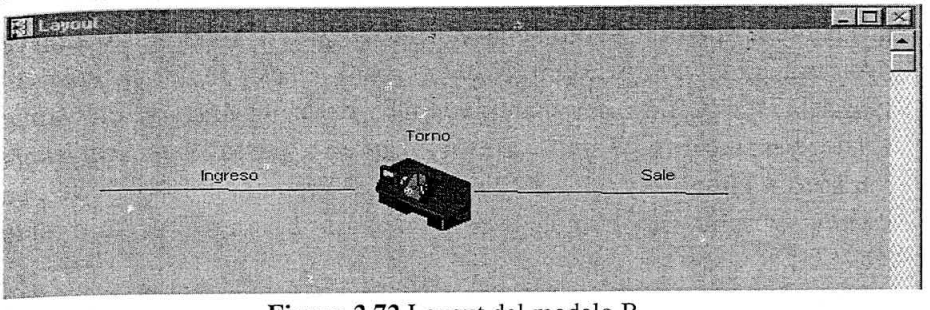


Figura 2.72 Layout del modelo B.

6. *BUILD (Locations)*

Locations

Icon	Name	Cap.	Units	DTs...	Stats...	Rules...	Notes...
	Ingreso	INFINI	1		Time Serie	Oldest, FIFO	
	Torno	1	1	None	Time Serie	Oldest	
	Sale	INFINI	1	None	Time Serie	Oldest, FIFO	

Figura 2.73 Instalaciones del modelo B.

7. *BUILD (Entities)*

Entities

Icon	Name	Speed (mpm)	Stats...	Notes...
	Subprod	50	Time Series	

Figura 2.74 Entidad del modelo B.

8. *BUILD (Arrivals)*

Arrivals

Entity...	Location...	Qty each...	First Time	Occurrences	Frequency	Logic	Di
Subprod	Ingreso	1		100	1		No

Figura 2.75 Llegadas del modelo B.

9. *BUILD (Processing)*

Entity	*Location*	*Operation*	*Output*	*Destination*	*Rule*	*Mv*
Subprod	Ingreso		Subprod	Torno		
Subprod	Torno	**WAIT** 5 **GRAPHIC** 2	Subprod	Sale		
Subprod	Sale		Subprod	EXIT		

10. *SIMULATION (Options)*

En la pantalla aparece el botón denominado *Browser*. Al oprimirlo aparece la ventana *Examinar para encontrar carpeta*. Hacer doble clic sobre el ícono del disco (C:), buscar el archivo *Temp* y oprimir *Aceptar*. En *Output path* debe aparecer "C:\Temp". Verificar que en el punto *Define run lenght by* aparezca *Time Only*, que el tiempo esté en minutos y los reportes, en *estándar*. Oprimir *Run*. El modelo debe comenzar a correr. Guardarlo como *MODELO B* y abrir de nuevo el MODELO A.

11. *FILE (Merge)*

Con el Modelo A instalado, ajustar la vista en 50% mediante *View-Zoom-50%*. Ir a *FILE-Merge*. Se abre la ventana *Model*. Al oprimir el botón, se abre una ventana que muestra los modelos; localizar allí el *MODELO B*. Cuando se selecciona el modelo B, aparece un aviso que le pregunta *¿en dónde quiere colocar el modelo?* Colocarlo a continuación del *MODELO A*. La *Salida* del *MODELO A* debe quedar muy cerca del *Ingreso* del *MODELO B*.

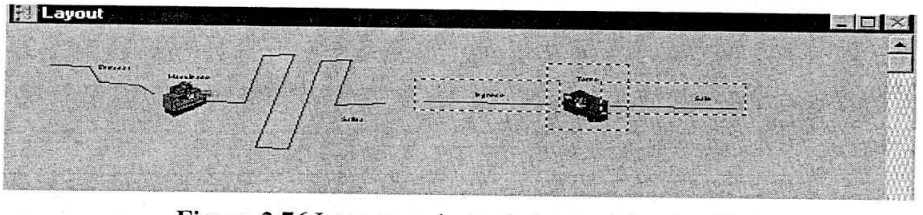


Figura 2.76 Layout conjunto de los modelos A y B.

12. *SIMULATION (Options)*

Mediante la opción *Save&Run, Guarde* este nuevo modelo con el nombre de *MEZCLA* y oprima *Run*.

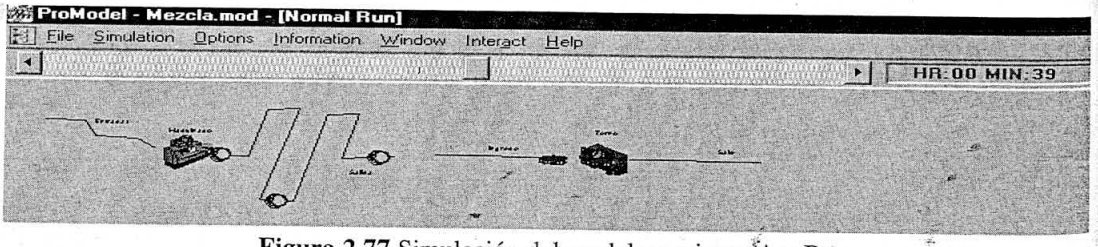


Figura 2.77 Simulación del modelo conjunto A y B.

El modelo *Mezcla* debe comenzar a funcionar.

Modelo 13. Uso de las instrucciones LOAD y UNLOAD

En este modelo se ilustra el uso de *LOAD* y *UNLOAD*, instrucciones de ProModel® que sirven para unir dos o más *Entidades*; por ejemplo *Ruedas* y *Pallets*. Aquí dos *Ruedas* se *cargan* sobre un *Pallet*, que recibe el nombre de *Pallet Cargado*; posteriormente se *descargan* y salen las *Ruedas* y los *Pallets* con su identidad inicial.

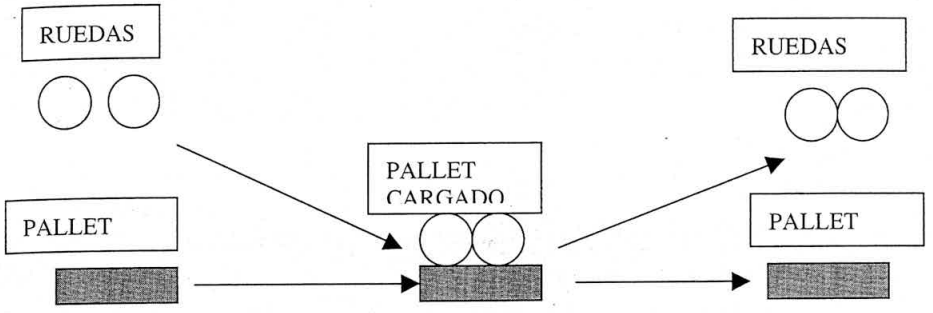


Figura 2.78 Ilustración del cargue y descargue de un pallet.

1. *BUILD (Locations)*

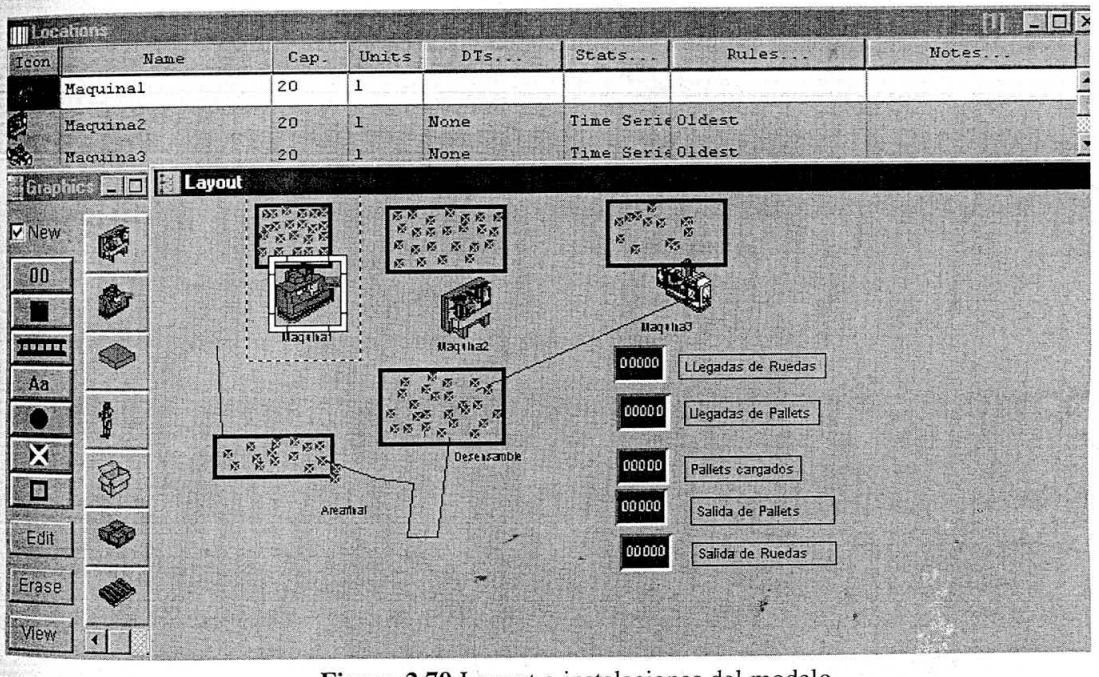


Figura 2.79 Layout e instalaciones del modelo.

En este caso, a las *Entidades* se les ha colocado un área de almacenamiento y, a algunas, una cola de salida. Para hacer esto, situarse en la ventana *Locations*, que debe aparecer iluminada, quitar el chulo de *New*, oprimir el cuadrado negro en la ventana *Graphics* y hacer clic en la ventana *Layout* en el sitio donde quiere colocar el almacenamiento, dentro del área punteada. Sin restablecer el chulo, colocar los cuadrados rojos con una X blanca de manera similar; después, oprimir el botón de la cola (parece una escalera horizontal) haciendo los quiebres con el botón izquierdo del *mouse* y soltando con el botón derecho. Hacer *doble* clic y definir como *Queue* (cola). Restablecer el chulo y continuar haciendo lo mismo para las demás *Instalaciones*.

2. *BUILD (Entities)*

Entities [1]

Icon	Name	Speed (mpm)	Stats...	Notes...
	Rueda	50	Time Series	
	Pallet	50	Time Series	
	Pallet_cargado	50	Time Series	

Figura 2.80 Entidades del modelo.

3. *BUILD(Arrivals)*

Entity...	Location...	Qty each...	First Time	Occurrences	Frequency	Logic	Disab
Rueda	Maquinal	1		2000	1		No
Pallet	Maquina2	1	0	500	1		No
Pallet_cargado	Maquina3	1	0	500	1		No

Figura 2.81 Llegadas al modelo.

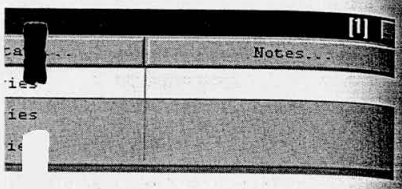
4. *BUILD (Attributes)*

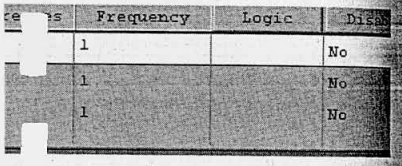
Attributes [3]

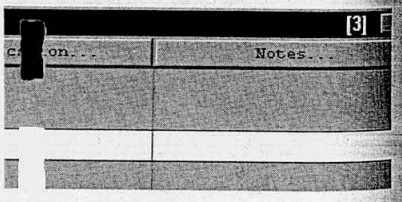
ID	Type...	Classification...	Notes...
Att1	Integer	Ent	
pallets	Integer	Ent	
ruedas		Ent	

Figura 2.82 Atributos del modelo.

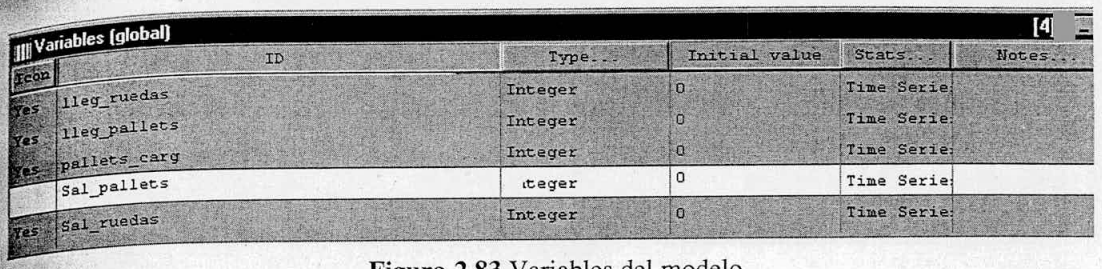
lo n área de almacena-
esto, situarse en la ven-
ır el chulo de *New*, oprí-
ıa :r clic en la ventana
amiento, dentro del área
d: dos rojos con una X
n : la cola (parece una
tón izquierdo del *mouse*
:) lefinir como *Queue*
) ismo para las demás







5. *BUILD (Variables)*


Variables [global] [4]

Icon	ID	Type...	Initial value	Stats...	Notes...
Yes	lleg_ruedas	Integer	0	Time Serie:	
Yes	lleg_pallets	Integer	0	Time Serie:	
Yes	pallets_carg	Integer	0	Time Serie:	
	Sal_pallets	Integer	0	Time Serie:	
Yes	Sal_ruedas	Integer	0	Time Serie:	

Figura 2.83 Variables del modelo.

6. *BUILD (Processing)*

Entity	*Location*	*Operation*	*Output*	*Destination*	*Rule*	*Move Logic*
Rueda	Máquina1	Att1 = 1 **WAIT** 2 **INC** lleg_ruedas ruedas = 1	Rueda	Máquina2	Load1	Move for 1
Pallet	Máquina2	**INC** lleg_pallets Att1 = 2 **LOAD** 2 pallets = 1	Pallet-cargado	Máquina3	Firtst1	Move for 1
Pallet-cargado	Máquina3	Att1 = 3 **WAIT** 10	Pallet-cargado	Desensamble	Firtst1	Move for 1
Pallet-cargado	Desensamble	**UNLOAD** 4 **IF** pallets = 1 **THEN** {**INC** Sal_pallets }	Pallet	Areafinal	Firtst1	Move for 1
Rueda	Desensamble		Rueda	Areafinal	Firtst1	Move for 1
Rueda	Areafinal	**WAIT** 10 **IF** ruedas = 1 **THEN** {**INC** Sal_ruedas }	Rueda	EXIT	Firtst1	Move for 1
Pallet	Areafinal		Pallet	EXIT	Firtst1	Move for 1

7. *SIMULATION (Options)*

En la pantalla aparece un botón llamado *Browser*. Al oprimirlo aparece la ventana *Examinar para encontrar carpeta*. Hacer doble clic sobre el ícono del disco (C:), buscar el archivo *Temp* y oprimir *Aceptar*. En *Output path* debe aparecer "C:\Temp". Verificar que en el punto *Define Run Lenght by* aparezca *Time Only*, que el tiempo esté en minutos y los reportes, en *estándar*. Oprimir *Run*.

Modelo 14. Pasteurizadora de leche

En este modelo, elaborado por José Alejandro Páez, se muestra cómo simular el proceso de pasteurización de la leche, que es un sistema de flujo continuo al comienzo y luego discreto, con las limitaciones obvias del ProModel® para el caso continuo.

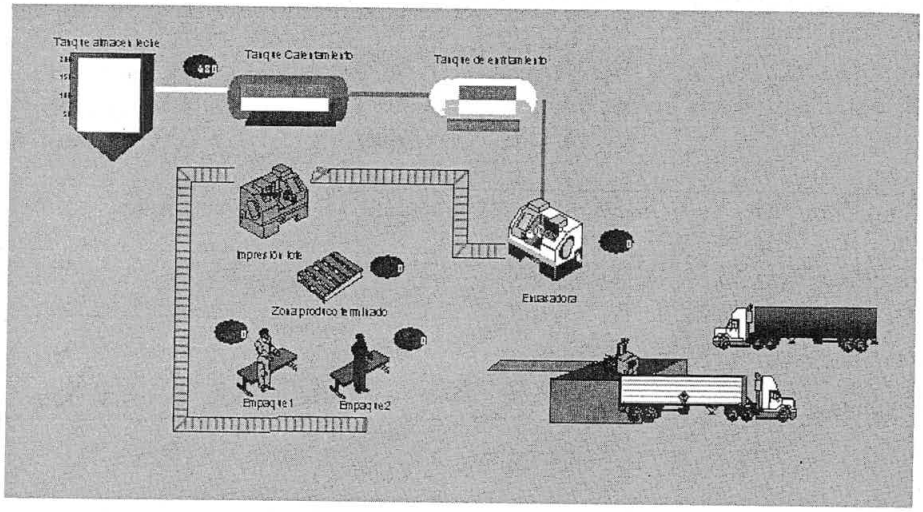


Figura 2.84 Layout de un proceso de pasteurización de leche.

1. *BUILD (Locations)*

Locations [15]

Icon	Name	Cap.	Units	DTs...	Stats...	Rules...	Notes...
	Tanque_almacen_leche	200	1	None	Time Series	Oldest, FIFO	En este tanque se rea
	Tuberia	20	1	None	Time Series	Oldest, FIFO	En la tuberia se real
	Tanque_Calentamiento	300	1	None	Time Series	Oldest, FIFO	En este tanque se cal
	Tuberia_caliente	70	1	None	Time Series	Oldest, FIFO	Esta tuberia transpor
	Tanque_de_enfriamient	200	1	None	Time Series	Oldest, FIFO	La leche es enfriada
	Tuberia_leche_past	50	1	None	Time Series	Oldest, FIFO	
	Envasadora	10	1	None	Time Series	Oldest, FIFO	Envasa la leche en ca
	Banda_transp	50	1	None	Time Series	Oldest, FIFO	
	Impresión_lote	1	1	None	Time Series	Oldest, FIFO	Imprime el lote y la
	Empaque1	20	1	None	Time Series	Oldest, FIFO	Empaca las cajas de l
	Empaque2	20	1	None	Time Series	Oldest, FIFO	Empaca las cajas de l
	Banda	50	1	None	Time Series	Oldest, FIFO	
	Zona_produco_terminad	100	1	None	Time Series	Oldest, FIFO	
	plataforma	100	1	None	Time Series	Oldest, FIFO	
	porteria	1	1	Jone	Time Series	Oldest	

Figura 2.85 Instalaciones del proceso de pasteurización.

El fondo de los tanques y las tuberías se dibujaron utilizando las herramientas que se encuentran en el menú *Build-Background Graphics-Front of Grid*. Sobre estos dibujos se colocaron "medidores de nivel". El medidor de nivel se encuentra en la segunda posición de la ventana *Graphics*.

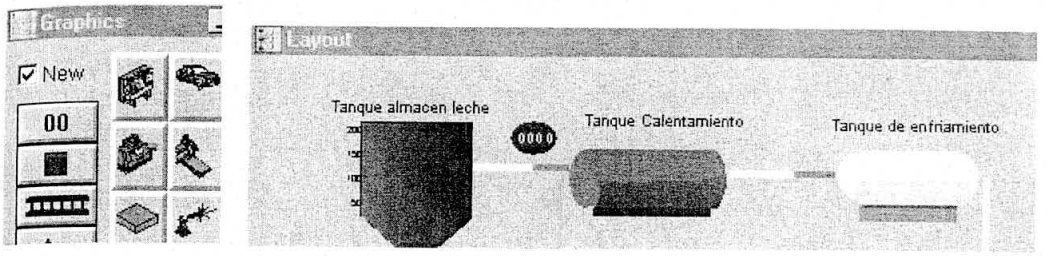


Figura 2.86 Figuras de fondo de los tanques y de la tubería sobre los que se colocan los medidores de nivel.

2. *BUILD (Entities)*

Entities [4]

Icon	Name	Speed (mpm)	Stats...	Notes...
	Leche	50	Time Series	
	Cajas_de_12	50	Time Series	
	Estiba	50	Time Series	

Figura 2.87 Entidades del modelo.

En el *Icono* de la entidad "leche" no se colocó ninguna figura para que el medidor de nivel "llene" los tanques y las tuberías a medida que la entidad avanza en el proceso de producción.

3. *BUILD (Arrivals)*

Arrivals [1]

Entity	Location	Qty each	First Time	Occurrences	Frequency	Logic	Disable
Leche	Tanque_almacen_1	1; Llegada_1	0	1000			No

Figura 2.88 Llegadas al sistema.

4. *BUILD (More Elements-Arrival Cycles)*

Arrival Cycles [1]

ID	Qty / $	Cumulative...	Table...
Llegada_leche		No	Defined

Table for Llegada_leche [9]

	Time (Hours)	Qty / $
1		2000
2		0
3		0
4		400
5		0
6		0
7		200
8		0

Figura 2.89 Ciclo de llegadas al sistema.

5. *BUILD (Attributes)*

Attributes [1]

ID	Type...	Classification...	Notes...
nivel_tanque	Integer	Ent	

Figura 2.90 Atributos del modelo.

6. *BUILD (Variables)*

Variables (global) [6]

Icon	ID	Type...	Initial value	Stats...	Notes...
Yes	Litros_leche	Integer	0	Time Series	
Yes	Num_cajas_de_leche	Integer	0	Time Series	
Yes	Num_cajas_con_lote1	Integer	0	Time Series	
Yes	Num_cajas_con_lote2	Integer	0	Time Series	
Yes	Num_cajas_de_12	Integer	0	Time Series	

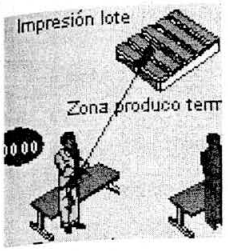
Figura 2.91 Variables del modelo.

7. *BUILD (Path Networks)*

Path Networks [4]

Graphic	Name	Type	T/S	Paths...	Interfaces...	Mapping...	Nodes
	Red1	Passing	Speed & Dista	1	2	0	2
	Red2	Passing	Speed & Dista	1	2	0	2
	Red3	Passing	Speed & Dista	1	2	0	2
	Red5		Speed & Dista	1	2	0	2

Figura 2.92 Redes del modelo.



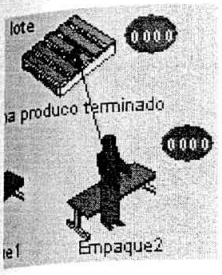
Paths [1]

From	To	3I	Distance
N1	N2	Bi	35.91

Interfaces [3]

Node	Location
N1	Empaque1
N2	Zona produco

Figura 2.93 Trayectoria e interfaces de la Red 1.



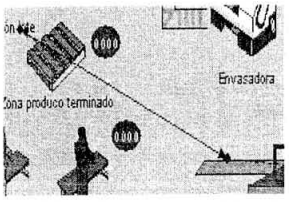
Paths [1]

From	To	3I	Distance
N1	N2	Bi	32.73

Interfaces [3]

Node	Location
N1	Empaque2
N2	Zona_produco_

Figura 2.94 Trayectoria e interfaces de la Red 2.



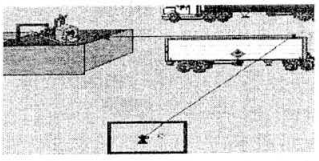
Paths [1]

From	To	3I	Distance
N1	N2	Bi	94.39

Interfaces [3]

Node	Location
N1	Zona_produco_
N2	plataforma

Figura 2.95 Trayectoria e interfaces de la Red 3.



Paths [1]

From	To	3I	Distance
N1	N2	Bi	139.06

Interfaces [3]

Node	Location
N1	plataforma
N2	portería

Figura 2.96 Trayectoria e interfaces de la Red 5.

8. *BUILD (Resources)*

Resources [4]

Icon	Name	Units	DTs...	Stats...	Specs...	Search...	Logic...	Pts...	Notes...
	Empacador1	1	None	By Unit	Red1, N1, 1	None	0	1	
	Empacador2	1	None	By Unit	Red2, N1, 1	None	0	1	
	Montacargas	1	None	By Unit	Red3, N2, 1	None	0	1	
	camión	1		Unit	Red5, N1, 1	None	0	1	

Figura 2.97 Recursos del modelo.

9. *BUILD (Processing)*

Entity	*Location*	*Operation*	*Output*	*Destination*	*Rule*	*Mv*
Leche	Tanque_almacén_leche	**INC** Litros-leche **WAIT** 5	Leche	Tubería	First 1	Move For 0.8
Leche	Tubería	**WAIT** 2	Leche	Tanque_calentamiento	First 1	Move For 0.8
Leche	Tanque_calentamiento	**WAIT** 20	Leche	Tubería_caliente	First 1	Move For 0.8
Leche	Tubería_caliente		Leche	Tanque_enfriamiento	First 1	Move For 0.8
Leche	Tanque_enfriamiento	**WAIT** 15	Leche	Tubería_leche_past	First 1	Move For 0.8
Leche	Tubería_leche_past		Leche	Envasadora	First 1	Move For 0.8
Leche	Envasadora	**WAIT** 2 **INC** Num_cajas_de_leche **GRAPHIC** 2	Leche	Banda_transp	First 1	Move For 0.8
Leche	Banda_transp		Leche	Impresión_lote	First 1	Move For 0.8
Leche	Impresión_lote	**WAIT** 0.5	Leche	Banda	First 1	Move For 0.8
Leche	Banda		Leche	Empaque 1 Empaque 2	0.5 0.5	Move For 0.8 Move For 0.8
Leche	Empaque 1	**WAIT** 2 **INC** Num_cajas_con_lote1 **ACCUM** 12 **GROUP** 12 **AS** Cajas_de_12				
Cajas_de_12	Empaque 1	**INC** Num_cajas_de_12	Cajas_de_12	Zona_producto_terminado	First 1	**MOVE WITH** Empacador 1 **THEN FREE**
Leche	Empaque 2	**WAIT** 2 **INC** Num_cajas_con_lote2 **ACCUM** 12 **GROUP** 12 **AS** Cajas_de_12				
Cajas_de_12	Empaque 2	**INC** Num_cajas_de_12	Cajas_de_12	Zona_producto_terminado	First 1	**MOVE WITH** Empacador 2 **THEN FREE**
Cajas_de_12	Zona_producto_terminado	**ACCUM** 10 **GROUP** 10 **AS** Estiba **WAIT** 2				
Estiba	Zona_producto_terminado	**WAIT** 2	Estiba	Plataforma	First 1	**MOVE WITH** Montacargas **FOR** 2 **THEN FREE**
Estiba	Plataforma	**ACCUM** 2	Estiba	Portería	First 1	**MOVE WITH** camión **FOR** 1 **THEN FREE**
Estiba	Portería		Estiba	EXIT	First 1	

stination	*Rule*	*Mv*
er	First 1	Move For 0.8
que_calenta nto	First 1	Move For 0.8
er en	First 1	Move For 0.8
que_enfria nto	First 1	Move For 0.8
er eche_	First 1	Move For 0.8
as ra	First 1	Move For 0.8
la transp	First 1	Move For 0.8
resion_lote	First 1	Move For 0.8
la	First 1	Move For 0.8
ac 1	0.5	Move For 0.8
aque 2	0.5	Move For 0.8
I ducto iii o	First 1	**MOVE WITH** Empacador 1 **THEN FREE**
p ucto ir)	First 1	**MOVE WITH** Empacador 2 **THEN FREE**
)1	First 1	**MOVE WITH** Montacargas **FOR** 2 **THEN FREE**
a	First 1	**MOVE WITH** camión **FOR** 1 **THEN FREE**
	First 1	

10. *SIMULATION (Options)*

En la pantalla aparece un botón llamado *Browser*. Al oprimirlo aparece la ventana *Examinar para encontrar carpeta*. Hacer un doble clic sobre el icono del disco (C:), buscar el archivo *Temp* y oprimir *Aceptar*. En *Output path* debe aparecer "C:\Temp". Verificar que en el punto *Define run lenght by* aparezca *Time Only*, que el tiempo esté en minutos, y los reportes, en *estándar*, *Run Hours* 8. Oprimir *Run*. El modelo debe comenzar a correr.

3

Casos de producción

Caso 1. Teoría de restricciones

Conceptos de *La meta* y *Factory physics*

Juan Carlos Rodríguez, ingeniero industrial recién egresado, ha comenzado a trabajar en Industrias ECI y ha sido asignado al área de producción con la tarea específica de mejorar la productividad. Juan Carlos todavía recuerda toda una serie de herramientas que aprendió en su curso de producción, relacionadas con dicha área: pronósticos, inventarios, MRP, JIT, PERT, planeación agregada, secuenciación, etc., pero no sabe por dónde comenzar.

En su última visita a la planta encontró exceso de inventarios, desorden, pedidos atrasados, órdenes de producción incompletas y un clima organizacional bastante caldeado, debido a las recriminaciones mutuas entre trabajadores y directivos.

Al comentar la situación con uno de sus compañeros de estudio, éste le recomendó que leyera *La meta*, libro de Eliyahu Goldratt[1], porque la situación que se describía correspondía a la narrada en dicha obra.

Parámetros de producción según Goldratt

Después de leer la obra, Juan Carlos comprendió que la meta de su empresa era "ganar dinero" y que para hacerlo debía entender muy bien el funcionamiento de nuevos parámetros de producción, como los siguientes:

***Throughput* (TH)**: Tasa a la cual la empresa produce unidades vendibles por unidad de tiempo (algunos traducen *Throughput* como facturación).

1. Eliyahu Goldratt y Jeff Cox, *La meta, un proceso de mejora continua*, 2a. ed., Monterrey, México, Ediciones Castillo.

***Work in Process* (WIP):** Cantidad de materias primas y de productos en proceso que se encuentran en el sistema.

WIP = Inventario en proceso en el sistema.

***Cycle Time* (CT):** Tiempo que gasta un producto desde que ingresa al sistema hasta que sale.

CT = Tiempo de ciclo.

Cuello de botella (CB): Operación más lenta del proceso, o la que produce el menor *Throughput* en determinado tiempo. Es la operación que marca el ritmo de la producción.

Tasa del cuello de botella (rb): Tasa de producción de la operación cuello de botella. Se mide en productos por unidad de tiempo.

Inventario: Todo el dinero que la empresa invierte en materiales, instalaciones, tecnología etc., que posteriormente se pueden vender.

Gastos de operación: Dinero que la empresa invierte para convertir los inventarios en *Throughput*.

Poco a poco, Juan Carlos comenzó a comprender que las actividades principales de organización de la producción debían encaminarse a incrementar el *Throughput*.

Para el gerente era inadmisible encontrar a un empleado descansando o a una máquina parada, razón por la cual los operarios permanecían siempre ocupados la mayor parte del tiempo, "adelantando trabajos". Juan Carlos comprendió que esta política únicamente conducía a incrementar los inventarios en proceso (*WIP*), a aumentar los gastos de operación, a congestionar el sistema productivo, pero no ayudaba a incrementar el *Throughput*. Pero, ¿cómo hacer ver al gerente que estaba equivocado?

Importancia de la simulación para solucionar el problema

Muy pronto, Juan Carlos recordó que en situaciones en las que la experimentación real es muy costosa, lo mejor es utilizar la simulación para observar diferentes comportamientos del sistema. La simulación es una representación ficticia de una situación real, que permite cambiar sus parámetros y variables. Se experimenta con un modelo, que es una abstracción de la realidad, y el conocimiento adquirido en la simulación se aplica en el mundo real. Cuanto mejor sea el grado de aproximación de la simulación a la realidad, mayor será su utilidad. El primer paso, entonces, era

construir el modelo del sistema, comenzando por dar respuesta a los siguientes interrogantes:

- ¿Cómo está distribuida la planta (*layout*)? ¿Cuáles son sus dimensiones?
- ¿Cuántos centros de trabajo (*locations*) tiene el proceso?
- ¿Cuáles son los tiempos de proceso de cada centro de trabajo?
- ¿Cuántos productos o subproductos (*entities*) se quieren incluir en el modelo?
- ¿Quién, dónde y cuándo requiere un recurso?
- ¿Qué se desea medir?
- ¿Cuáles serán los indicadores que se deberán tener en cuenta?
- ¿Cuánto tiempo se simulará el proceso?
- ¿Qué tan variable es el proceso? ¿Se podría estandarizar?
- ¿Quién es la persona que utilizará el programa de simulación?

Parámetros y leyes de la producción según Hopp y Spearman

En su proceso de investigación bibliográfica, Juan Carlos encontró el libro *Factory physics*[2], también relacionado con las ideas de Goldratt. En dicho libro se definen algunos parámetros y leyes importantes de la *física de planta*:

Tiempo de proceso global (To): Suma de los tiempos de proceso promedio de todas y cada una de las estaciones de trabajo, a largo plazo. Si *ti* es el tiempo de proceso de la estación *i*, entonces:

$$To = \Sigma ti, \text{ para toda estación de trabajo } i.$$

Tasa del Cuello de Botella (rb): Tasa o rata (el número de partes o trabajos por unidad de tiempo) de producción de la estación de trabajo u operación que tiene la *mínima capacidad* del proceso. Si *ri* es la tasa de producción de la operación *i*, entonces:

$$rb = min\ (ri), \text{ para toda operación } i.$$

2 Wallace J. Hoppe y Mark L. Spearman, *Factory Physics, Foundations of Manufacturing Management*, cap. 7, Boston, Irwin-McGraw-Hill, Inc., 1996.

WIP crítico de la línea (Wo): Nivel de inventario con el que la línea produce el máximo *Throughput*. A partir de dicho nivel, el inventario únicamente produce congestión y costos de almacenamiento.

$$Wo = (rb).To$$

Ley de Little: $TH = (WIP/CT)$

La ley de Little establece que hay una relación estrecha entre el *Throughput* (*TH*), el *Cycle Time* (*CT*) y el *Work In Process* (*WIP*), definida por la fórmula anterior, en la que se observa que cuanto más largo sea el tiempo de ciclo, menor será el *Throughput* con un nivel de inventario constante.

Ley del mejor desempeño:

$$TH\text{mejor} = w/To \text{ si } w < Wo$$

THmejor $= rb$ si $w > Wo$, o $w = Wo$, siendo w el nivel de *WIP*

La ley del mejor desempeño establece que para un nivel dado de inventario w, el mayor *Throughput*, r_b, se obtiene cuando el nivel de inventario es igual o mayor que el nivel crítico Wo, y es igual a r_b, o sea la tasa del cuello de botella.

La ley del mejor desempeño también se puede formular de la siguiente manera:

$$CT\text{mejor} = To \text{ si } w < Wo \text{ o } w = Wo$$

$$CT\text{mejor} = w/rb, \text{ si } w > Wo$$

El mejor tiempo de ciclo es el que corresponde al nivel crítico de inventario *Wo*.

Ley del peor desempeño práctico:

THpeor $= (w/(Wo+w-1))rb$, para cualquier nivel de inventario w.

CTpeor $= To + (w-1)/rb$, para cualquier nivel de inventario w.

Juan Carlos encontró además un esquema[3] que relaciona los parámetros y variables vistos con la variabilidad:

3 Alejandro Salazar, *Lecciones de física de planta-Conwip*, Video Top Management.

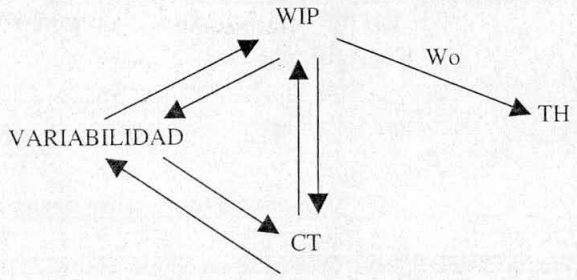


Figura 3.1 Relación de la variabilidad con el *Throughput, el WIP y el Cycle time.*

Al incrementar la variabilidad, se aumentan el tiempo de ciclo y el inventario en proceso, empeorándose el *Throughput*. Además, si se incrementa el *WIP*, se alarga el tiempo de ciclo y, muy probablemente, se disminuye el *Throughput.*

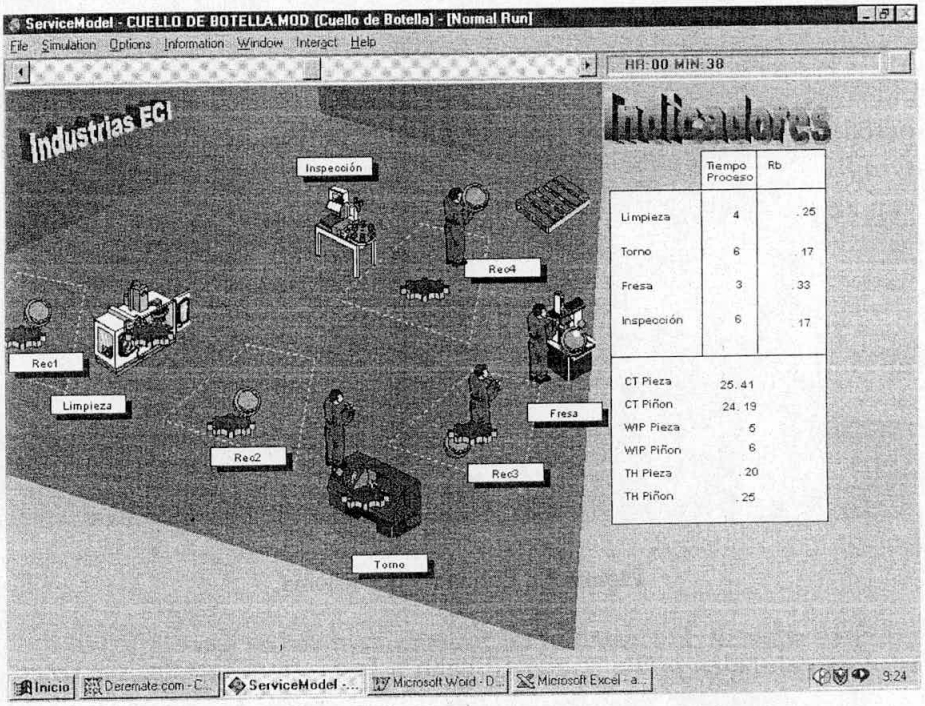


Figura 3.2 Distribución de la planta (*layout*) de Industrias ECI.

Modelando el sistema de producción de Industrias ECI

Para convencer al gerente, Juan Carlos elaboró un modelo con nueve instalaciones (*locations*): *limpieza*, *torno*, *fresa*, *inspección*, *paletizadora*, *re-*

*cepción*1, *recepción*2, *recepción*3, *recepción*4. Los datos iniciales fueron los siguientes:

1. *BUILD (Locations)*

Locations [10]

Icon	Name	Cap.	Units	DTs...	Stats...	Rules...	Notes...
	Limpieza	1	1	None	Time Serie:	Oldest	
	Torno	1	1	None	Time Serie:	Oldest	
	Fresa	1	1	None	Time Serie:	Oldest	
	Inspección	1	1	None	Time Serie:	Oldest	
	Pallet_	inf	1	None	Time Serie:	Oldest	
	Rec1	inf	1	None	Time Serie:	Oldest	
	Rec2	inf	1	None	Time Serie:	Oldest	
	Rec3	inf	1	None	Time Serie:	Oldest	
	Rec4	inf	1	None	Time Serie:	Oldest	

Figura 3.3 Instalaciones del modelo.

Las recepciones se concibieron como áreas marcadas en el piso y capacidad teórica de almacenamiento infinita. El letrero "Industrias ECI", las paredes y el piso se construyeron mediante el menú *Build-Background Graphics-Front Of Grid* usando las herramientas de dibujo que aparecen allí.

2. *BUILD (Entities)*

En el modelo se incluyeron dos entidades: *piñón* y *pieza*.

Entities [3]

Icon	Name	Speed (mpm)	Stats...	Notes...
	Pinon	50	Time Series	
	Pieza	50	Time Series	

Figura 3.4 Entidades del sistema.

Las entidades *piñón* y *pieza* llegan a la primera instalación *recepción*1.

3. *BUILD (Arrivals)*

Los *piñones* están llegando con una frecuencia de uno cada 5 minutos y las *piezas* con una frecuencia de una cada 7 minutos. Se generan 100 ocurren-

cias; esto quiere decir que durante el tiempo de simulación, estarán llegando 200 entidades al sistema.

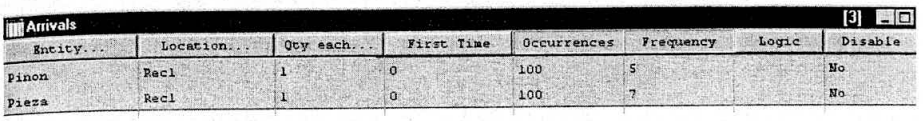


Figura 3.5 Llegadas al sistema.

4. *BUILD (Path Network)*

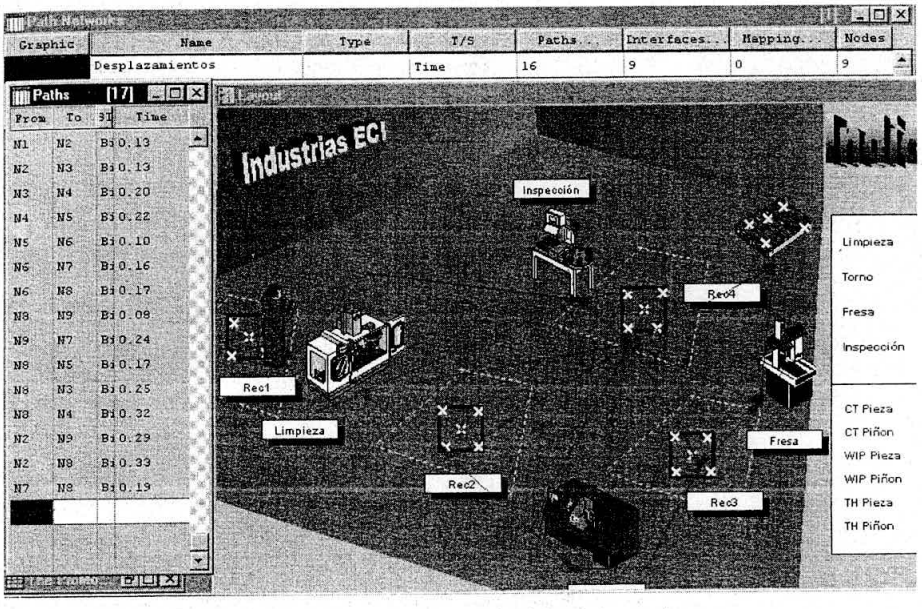


Figura 3.6 Definición de la red *Desplazamientos*.

Se definió una red de movimientos o *Path Network*, llamada desplazamientos, con 16 *trayectorias* o *Paths bidireccionales*, haciendo coincidir cada una de las *locations* con un *nodo de la red*, mediante las interconexiones o *interfaces*.

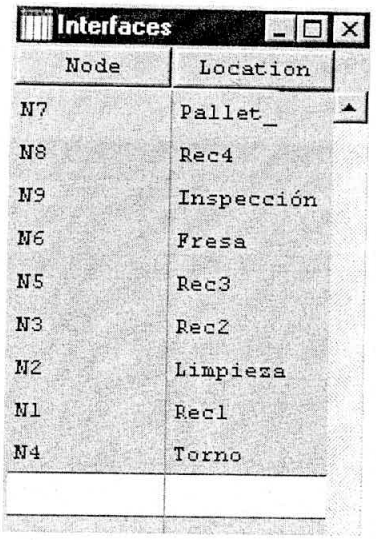


Figura 3.7 Interfaces de la red de movimientos *Desplazamientos.*

5. BUILD (Resources)

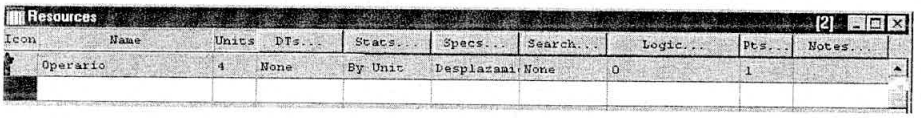


Figura 3.8 Recursos del modelo.

Al hacer clic en *Specifications* aparecerá una ventana con dicho nombre. Allí se define *Path nethwork* como *desplazamientos*, *Home* como N1 y se marca *Closest Resource*. Esta información dice al modelo que los operarios se mueven de acuerdo con las trayectorias de la red *desplazamientos*, definida previamente; que el sitio de trabajo de los operarios es el *Nodo 1*, o sea, recepción1, adonde se dirigirán cuando no estén ocupados, y que se debe utilizar el *operario que esté más cercano* a la instalación donde se le requiera. Las unidades se definen como 4 y las demás se ajustan automáticamente.

Al hacer clic en *Points* aparecerá una ventana donde se definen las coordenadas al *Nodo 1* como *(0,0)*. Se puede "arrastrar el Nodo", si no coinciden; esto permitirá al sistema ubicar cartesianamente todas las instalaciones del modelo.

6. *BUILD (Attributes)*

Attributes [4]

ID	Type...	Classification...	Notes...
att_pinon	Integer	Ent	
att_pieza	Integer	Ent	
att_general	Integer	Ent	

Figura 3.9 Atributos del sistema.

Se utilizaron tres *atributos* que son "super-variables", en donde se guarda, en este caso, "el tiempo de llegada al sistema" de cada una de las entidades. Obsérvese que se usa la palabra *general* cuando las dos entidades *piñón* y *pieza* entran simultáneamente en el sistema.

7. *BUILD (Variables)*

Variables (global) [1

Icon	ID	Type...	Initial value	Stats...	Notes
Yes	Rb1	Real	0	Time Serie	
Yes	Rb2	Real	0	Time Serie	
Yes	Rb3	Real	0	Time Serie	
Yes	Rb4	Real	0	Time Serie	
Yes	tp_1	Integer	0	Time Serie	
Yes	tp_2	Integer	0	Time Serie	
Yes	tp_3	Integer	0	Time Serie	
Yes	tp_4	Integer	0	Time Serie	
Yes	CT_pinon	Real	0	Time Serie	
Yes	CT_pieza	Real	0	Time Serie	
Yes	wip_pinon	Integer	0	Time Serie	
Yes	wip_pieza	Integer	0	Time Serie	
Yes	TH_Pieza	Real	0	Time Serie	
Yes	TH_pinon	Real	0	Time Serie	
No	th_general	Real	0	Time Serie	
No	wip_general	Integer	0	Time Serie	
No	ct_general	Integer	0	Time Serie	

Figura 3.10 Variables del sistema.

Se utilizaron 17 variables para medir tasas de producción (*ri*), tasa del cuello de botella (*rb*), tiempos de proceso (*tp*), tiempos de ciclo (*ct*), inventarios en proceso (*wip*) y *throughput* (*th*).

8. *BUILD (Subroutines)*

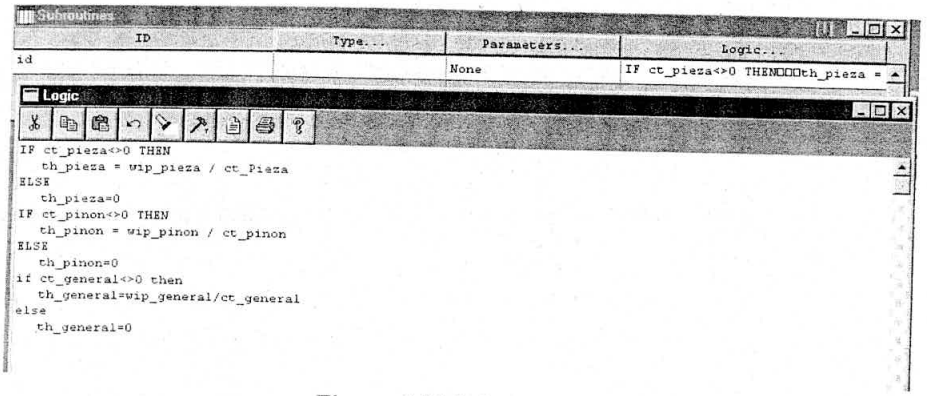


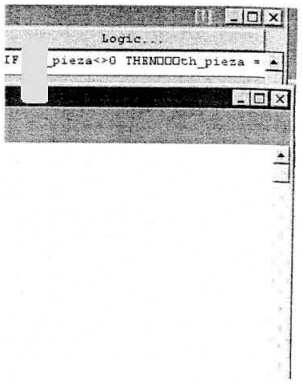
Figura 3.11 Subrutina *id.*

Esta *subrutina* hace lo siguiente: si el tiempo de ciclo es diferente de cero, calcula el *Throughput* dividiendo el *Wip* entre el *tiempo de ciclo*; de lo contrario, hace el *Throughput* igual a cero. Su papel es evitar que en el proceso aparezca una división entre cero, debido a que el tiempo de ciclo sólo se calcula cuando haya salido la primera pieza.

9. *BUILD (Processing)*

Como el proceso es un poco largo, se irá comentando en grupos de 4 instrucciones:

Entity	Location	Operation	Output	Destination	Rule	Move Logic
Piñon	Recepción1	**REAL** x id att_piñón = **CLOCK**() **INC** Wip_piñón **INC** Wip-general	Piñon	Limpieza	First 1	**MOVE WITH** Operario **THEN** **FREE**
Piñon	Limpieza	**REAL** x id Tp1=4 x= n (Tp1, .8) Rb1= 1/x **WAIT** x	Piñon	Recepción2	First 1	**MOVE WITH** Operario **THEN** **FREE**
Piñon	Recepción2	**REAL** x id	Piñon	Torno	First 1	**MOVE WITH** Operario **THEN** **FREE**
Piñon	Torno	**REAL** x id Tp2=6 x=n(Tp2,1) Rb2=1/x **WAIT** x	Piñon	Recepción3	First 1	**MOVE WITH** Operario **THEN** **FREE**



ıdo en grupos de 4 ins-

Rule	Move Logic
F t 1	**MOVE WITH** Operario **THEN FREE**
F 1	**MOVE WITH** Operario **THEN FREE**
First 1	**MOVE WITH** Operario **THEN FREE**
F 1	**MOVE WITH** Operario **THEN FREE**

En la primera instrucción, la entidad *piñón* llega a la instalación recepción1 e inicia la siguiente operación: mediante *REAL x*, el programa crea una *variable interna x* donde va a almacenar un tiempo de operación; la subrutina *id* chequea si los tiempos de ciclo del piñón, de la pieza y el tiempo general son diferentes de cero (*Ct_piñon, Ct_pieza, Ct_general*). Luego, el programa coloca la hora que está marcando el reloj de la simulación en el momento en que entra la entidad *piñón* al sistema, en el atributo *att_piñón* (es el mismo efecto de una persona que marca tarjeta en el reloj de la entrada). Después, el programa incrementa las variables *Wip_piñón* y *Wip-general*, donde se está guardando la información de los inventarios en proceso de *piñones* y la *general* de *piñones* y *piezas*.

El efecto de las instrucciones *First 1* y *MOVE WITH Operario THEN FREE* es el siguiente: la primera entidad *piñón* que se encuentre disponible pasa a la instalación *limpieza*, mediante uno de los cuatro *operarios*, quien recoge el *piñón*, lo lleva hasta *limpieza* y queda libre, para retornar a su sitio de trabajo en el *Nodo1*. Estas dos últimas instrucciones se repiten de aquí en adelante, de manera que no se comentarán más.

En la segunda instrucción, mediante *REAL x*, el programa crea una *variable interna x* en donde va a almacenar un tiempo definido por una distribución normal con media *Tp1* y desviación estándar 0.8. En el paso anterior se almacena en *Tp1* el valor 4, que corresponde al tiempo de proceso de la *recepción1*. A continuación, en la variable *Rb1* que es la tasa del cuello de botella de la *limpieza* se guarda el valor *1/x*. Con *WAIT x* el programa hace que la entidad *piñón* espere un tiempo *n(4,0.8)* en la operación de *limpieza*.

En la tercera instrucción, la entidad *piñón*, es llevada desde la *recepción2* hasta el *torno* por uno de los cuatro operarios.

La cuarta instrucción es similar a la segunda.

Las siguientes cuatro instrucciones son:

Entity	Location	Operation	Output	Destination	Rule	Move Logic
Piñon	Recepción3	**REAL** x id	Piñon	Fresa	First 1	**MOVE WITH** Operario **THEN** **FREE**
Piñon	Fresa	**REAL** x Id Tp3=5 x= n (Tp3, .4) Rb3= 1/x **WAIT** x	Piñon	Recepción4	First 1	**MOVE WITH** Operario **THEN** **FREE**
Piñon	Recepción4	**REAL** x id	Piñon	Inspección	First 1	**MOVE WITH** Operario **THEN** **FREE**
Piñon	Inspección	**REAL** x id Tp4=2 x=n(Tp4,.2) Rb4=1/x **WAIT** x	Piñon	Paletizadora	First 1	**MOVE WITH** Operario **THEN** **FREE**

Este grupo de instrucciones es similar a las del grupo anterior. Las cuatro siguientes son:

Entity	Location	Operation	Output	Destination	Rule	Move Logic
Piñon	Paletizadora	**REAL** x id **DEC** Wip_piñón **DEC** Wip_general	Piñon	EXIT	First 1	Ct_piñon= **CLOCK()**-att_piñón Ct_general=Ct_piñón
Pieza	Recepción1	**REAL** x id att_pieza=**CLOCK()** **INC** Wip_pieza **INC Wip_general** **WAIT** x	Pieza	Limpieza	First 1	**MOVE WITH** Operario **THEN** **FREE**
Pieza	Limpieza	**REAL** x id Tp1=2 x=n(Tp1,.5) Rb1=1/x **WAIT** x	Pieza	Recepción2	First 1	**MOVE WITH** Operario **THEN** **FREE**
Pieza	Recepción2	**REAL** x id	Pieza	Fresa	First 1	**MOVE WITH** Operario **THEN** **FREE**

En este grupo de instrucciones se observa:

En la primera instrucción, la entidad *piñón*, luego de pasar por la *paletizadora* abandona el sistema, de manera que deja de ser inventario en proceso, para convertirse en producto terminado y, por lo tanto, en *Throughput*. Los inventarios en proceso *Wip_piñón* y *Wip_general* se

ıle	Move Logic
st 1	**MOVE WITH** Operario **THEN** **FREE**
st 1	**MOVE WITH** Operario **THEN** **FREE**
First 1	**MOVE WITH** Operario **THEN** **FREE**
First 1	**MOVE WITH** Operario **THEN** **FREE**

upo anterior. Las cua-

Rule	Move Logic
First 1	Ct_piñon= **CLOCK**()-att_piñón Ct_general=Ct_piñón
First 1	**MOVE WITH** Operario **THEN** **FREE**
First 1	**MOVE WITH** Operario **THEN** **FREE**
F 1	**MOVE WITH** Operario **THEN** **FREE**

ego de pasar por la
ser inventario en
por lo tanto, en
y *Wip_general* se

decrementan. El *tiempo de ciclo* del *piñón* se calcula restando, del tiempo actual de la simulación (que debe estar marcando el momento en que *piñón* sale del sistema), el tiempo de entrada guardado en el atributo *att_piñón*. A continuación, se toma como tiempo de ciclo general el tiempo de ciclo del *piñón*. Las demás instrucciones de este grupo son similares a las anteriores.

Las siguientes cuatro instrucciones son:

Entity	Location	Operation	Output	Destination	Rule	Move Logic
Pieza	Fresa	**REAL** x id Tp2=3 x=n(Tp2,.6) Rb2=1/x **WAIT** x	Pieza	Recepción3	First 1	**MOVE WITH** Operario **THEN** **FREE**
Pieza	Recepción3	**REAL** x Id	Pieza	Torno	First 1	**MOVE WITH** Operario **THEN** **FREE**
Pieza	Torno	**REAL** x Id Tp3=4 x=n(Tp3,1) Rb3=1/x **WAIT** x	Pieza	Recepción4	First 1	**MOVE WITH** Operario **THEN** **FREE**
Pieza	Recepción4	**REAL** x id	Pieza	Inspección	First 1	**MOVE WITH** Operario **THEN** **FREE**

En este grupo las instrucciones son similares a las anteriores. Las dos últimas instrucciones son:

Entity	Location	Operation	Output	Destination	Rule	Move Logic
Pieza	Inspección	**REAL** x id Tp4=6 x=n(Tp4,.2) Rb4=1/x **WAIT** x	Pieza	Paletizadora	First 1	**MOVE WITH** Operario **THEN** **FREE**
Pieza	Paletizadora	**REAL** x id **DEC** Wip_pieza **DEC** Wip_general	Pieza	EXIT	First 1	Ct_pieza= **CLOCK**()-att_pieza Ct_general=Ct_pieza

Estas instrucciones también son similares a las anteriores.

10. *SIMULATION (Options)*

No olvidar colocar en *Output path: C:\Temp; Run hours: 8; Output reporting: Standard.*

Análisis de los resultados del modelo

El modelo se corrió al principio durante 8 horas sólo con la entidad *piñón*, deshabilitando la entidad *pieza*. Para deshabilitar, ir a *Arrivals-Disable* colocando *Yes* en la entidad que no se quiere tener en cuenta. Los siguientes son datos obtenidos con las dos entidades, entrando en el sistema:

Entidad	Tiempo de Proceso	Tasa del cuello de botella rb
Limpieza	4	.25
Torno	6	.16
Fresa	3	.30
Inspección	2	.50

Ct_pieza: 240 minutos

Ct_piñón: 236.13 minutos

Wip pieza: 34

Wip piñón: 49

Th pieza: .14 pieza/hora

Th piñón: .2 piñón/hora

Como se observa, si las dos entidades entran en el sistema, el *cuello de botella* lo constituye el *torno*. Como se simuló un sistema *Push*, en el cual se trata de trabajar continuamente, buscando obtener el máximo *Throughput*, se observa que el *WIP* (inventario en proceso) se dispara a 34 partes de *pieza* y a 49 partes de *piñón*. Con ésta información, Juan Carlos puede mostrarle a su Gerente, como en estos sistemas se dispara el WIP.

Cuando el sistema trabaja con la entidad *piñón* únicamente, el *cuello de botella* es el *torno*, mientras que si produce *piezas* únicamente, el *cuello de botella* es la operación *limpieza*.

Después de correr el programa, ProModel® pregunta si quiere ver los resultados. Con el módulo *General Report* se puede observar el comporta-

miento de cada una de las *instalaciones*, de los *recursos*, de las *entidades*, de las *variables*, etc. En la parte superior del reporte se encuentra un menú de herramientas, en el cual se pueden distinguir cuatro íconos principales: *Barras Horizontales, Líneas, Barras Verticales y Una especie de cono.*

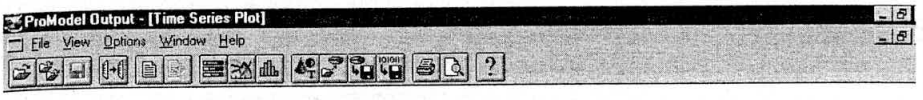


Figura 3.12 Barra de herramientas de los reportes.

Al hacer clic en el ícono de las líneas aparece la ventana *Chose Statistics For Plot* con una serie de variables o *values* que se marcan y se trasladan a la ventana del frente, *Selected Values*, con una flechita que está en el centro de las dos ventanas; luego se da *OK* y aparece la gráfica. Si se quiere colocar letreros, promedios, etc., se hace clic en el ícono del cono y aparece la ventana *Graph Options*, que tiene opción para colocar títulos en los ejes y para entrar en la ventana *Plot Options*, donde se pueden colocar los promedios y definir el tipo de gráfico que se requiere.

Juan Carlos sabe que la instalación que marca el ritmo de la producción es el *torno*, de manera que deberá preocuparse por que esta máquina esté funcionando permanentemente. Esto significa que habrá que programarle mantenimiento preventivo siempre, lo mismo que mantenimiento predictivo y correctivo. Además, él deberá asegurarse de que a esta máquina no le falten materiales y de que los que lleguen cumplan con los requisitos de calidad establecidos. Deberá también reducir su tiempo de ciclo, acortando los tiempos de alistamiento (montajes, lubricaciones, etc.)

Juan Carlos tiene ahora evidencia para recomendar a su gerente la implementación de un sistema de producción *pull* que controle el *WIP*, con el fin de mejorar la productividad.

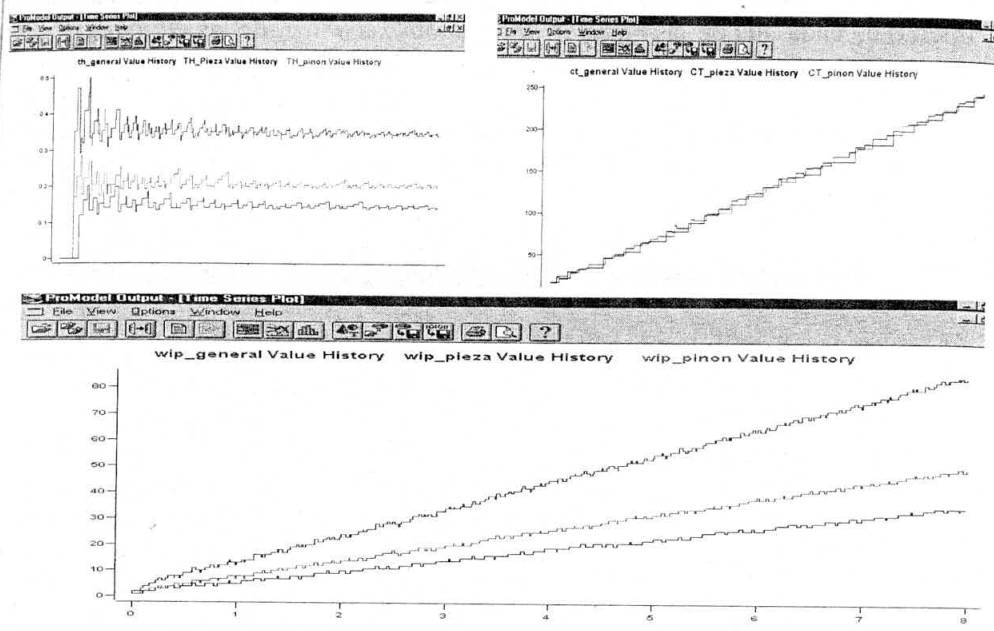


Figura 3.13 Gráficas ProModel del *WIP*, *Throughput* y *Cycle Time* para cada una de las entidades *pieza*, *piñón* y para las dos, simultáneamente (*general*), de un sistema *Push*. Obsérvese que, en este sistema, el *WIP* se dispara.

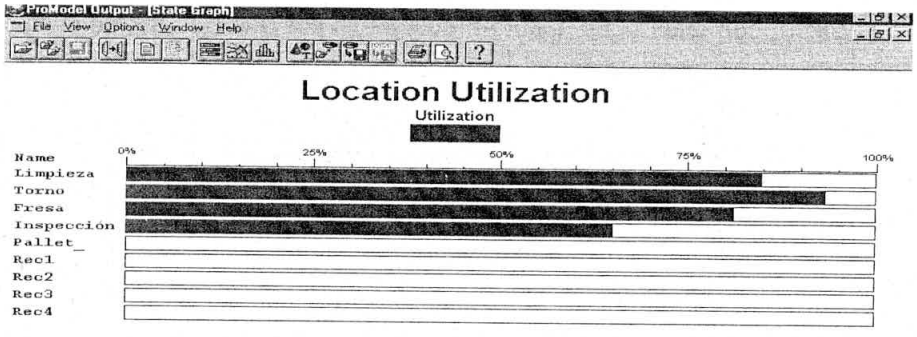
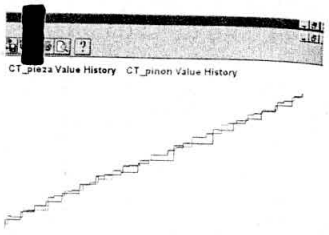
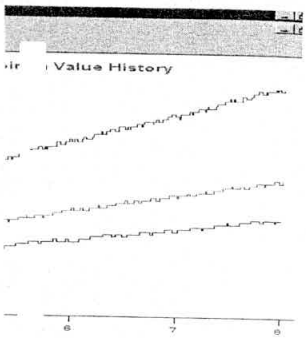
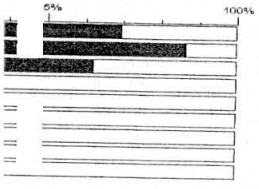


Figura 3.14 Utilización de instalaciones. Obsérvese que la operación cuello de botella es el *torno*.





y ? *Time* para cada una
n te (*general*), de un
WIP se dispara.



a peración cuello de

Caso 2. Cajero automático

Colas M/M/1

José Rodríguez, jefe del departamento de operaciones del Banco Urbano, desea calcular el costo promedio del servicio por cliente, y la eficiencia de un cajero automático, ubicado en la parte exterior del banco. José ha recolectado información del servicio del cajero automático, y al analizarla ha encontrado lo siguiente:

a) Tipos de llegadas de clientes

Se distinguen tres tipos de llegadas de clientes, que dependen de la hora del día:

1. **Alta demanda** (6 a.m. a 9 a.m. y 3 p.m. a 6 p.m.) En este horario diurno, el banco se encuentra cerrado y 40 clientes utilizan el cajero automático, en promedio.
2. **Mediana demanda** (9 a.m. a 3 p.m. y 6 p.m. a 9 p.m.) En este horario diurno, el banco se encuentra abierto y 20 clientes, en promedio, prefieren utilizar el cajero automático.
3. **Baja demanda** (9 p.m. a 6 a.m.) En este horario nocturno, el banco se encuentra cerrado y llegan 10 clientes en promedio.

b) Tiempo de servicio y costos

El tiempo promedio de servicio del cajero se ha estimado en dos minutos por cliente.

Si un cliente encuentra más de quince personas haciendo fila, abandona el sistema.

El costo fijo de funcionamiento del cajero se estima en $ 10.000 por hora.

El costo de oportunidad por cada cliente que abandona el cajero sin ser atendido se estima en $ 20.000 por cliente.

c) Información requerida

José necesita conocer:

- Número promedio de clientes diarios, discriminados por tipo de demanda, para cada una de las situaciones que se describen más adelante.
- Porcentaje diario de ocupación del cajero automático, discriminado por tipo de demanda, para cada una de las situaciones que se describen más adelante.
- Número de clientes diarios que no fueron atendidos por el cajero automático y su costo de oportunidad, para cada una de las situaciones que se describen más adelante.

d) Situaciones

1. *Tiempos discretos*

	Alta demanda	Mediana demanda	Baja demanda
Número de llegadas por hora	40	20	10
Tiempo de servicio (min.)	2	2	2

2. Tiempos aleatorios (Cola M/M/1)

	Alta demanda	Mediana demanda	Baja demanda
Número de llegadas por hora	Poisson (40)	Poisson (20)	Poisson (10)
Tiempo de servicio (min.)	Normal (2,0.5)	Normal (2,0.5)	Normal (2,0.5)

En qué consiste una cola M/M/1

Una cola M/M/1 es un sistema conformado por *un servidor*, en este caso el cajero automático, y una serie de *clientes* que llegan de manera aleatoria al servidor a solicitar un servicio.

Los clientes llegan al sistema de uno en uno. El *número de llegadas* de clientes, que suceden en un intervalo de tiempo t, es una variable aleatoria

independiente y discreta que se modela muy bien con una distribución de Poisson, con una *tasa entre llegadas* Lambda (L) y *número de llegadas* tL^4.

El servidor *atiende* a los clientes con una *duración de servicio*, otra variable aleatoria independiente y continua que se modela con una distribución exponencial negativa con *tasa de servicio* Mu (M). Tan pronto un cliente es servido, abandona el sistema.

Para que se forme cola al frente al servidor, *la tasa de llegada de clientes* debe ser mayor que *la tasa de servicio*, L > M.

Un cajero automático es un ejemplo de sistema de colas *M/M/1*. (La primera *M* recuerda que es un *proceso de Markov*, de *número de llegadas* Poisson; la segunda *M* informa que *el tiempo de servicio* es exponencial y el *1* que se tiene un solo servidor). Cualquier otro tipo de sistema de producción o de operaciones, con *tiempo de servicio* aleatorio, con llegadas aleatorias y con un solo servidor, se puede modelar como una cola *M/M/1*.

Modelando el cajero automático con ProModel

1. *BUILD (Locations)*

Se consideran tres instalaciones o *locations*: *cola_cajero*, *cajero* y *cola_salida*. Los datos iniciales son:

Name	Capacity	Units	Down Times	Statistics
Cola_cajero	15	1	None	Time Series
Cajero	1	1	None	Time Series
Cola_salida	Infinite	1	None	Time Series

La *cola_cajero*, la primera *location*, se construyó de la siguiente manera: En la ventana *Graphics* del menú *Build-Locations* hay tres elementos que se utilizaron así: el primero, es un contador que se coloca frente al letrero *Personas en cola*, el segundo, es una cola que al editarla con un clic, se definió como *Queue, Line, Invisible During Simulation* y el tercero, es un cuadrado rojo con una X blanca, que sirve para indicar la posición que ocupará la entidad que llega y se colocó en la cabeza de la cola. La capacidad de *cola_cajero* se limitó a quince personas, lo que hace que las entidades que lleguen cuando haya más de quince clientes, abandonen el sistema.

4 Wayne L. Winston, *Investigación de operaciones, aplicaciones y algoritmos*, México, Grupo Editorial Iberoamérica, 1994, p. 1114.

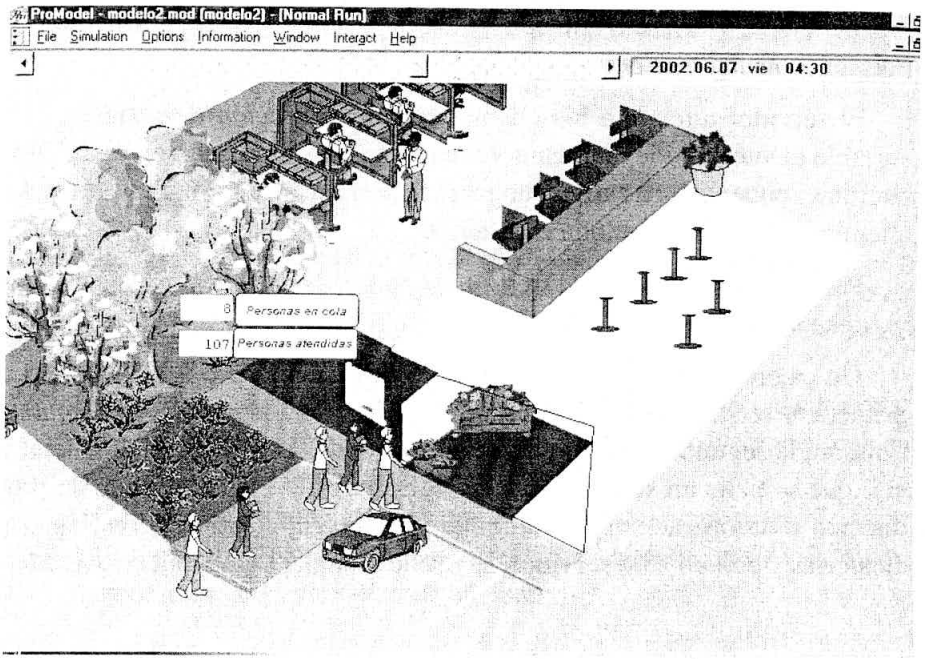


Figura 3.15 Distribución (*layout*) del cajero automático.

Los letreros *Personas en cola* y *Personas atendidas* se elaboran con *Build– Background Graphics–Front of Grid*, lo mismo que el fondo del *layout*. Los contadores que aparecen al frente de los letreros corresponden al ícono de la variable *num_persona* (personas atendidas), que se define más adelante y al de la *cola_cajero* (personas en cola).

2. *BUILD (Entities)*

Se trabaja con una sola entidad de flujo o *entity*, llamada *cliente*. En el momento de definir las entidades se ingresa como *gráfica 1* un hombre, luego se quita el chulo de *New* en la ventana de *Entity–Graphics* y se coloca como *gráfica 2* una mujer.

3. *BUILD (More elements-External files)*

Aquí se introduce la información de los nombres de los archivos externos, sus características y su ubicación.. El nombre *demanda* se define en *C: Temporal*, llamado *datos.txt*, y es un archivo de escritura.

Identification	Type	File Name
Cliente_atendidos	General Write	C:\Temp\ clientes_atendidos.txt
demanda	General Write	C: \Temp\ datos.txt

4. *BUILD (Arrivals)*

Las llegadas de clientes al sistema o *arrivals* se definen así:

Entity	Location	Quantity	First Time	Occurrences	Frequency	Logic
Cliente	Cola_cajero	1; ciclo_1	0	1	24 hr	Ver (*)

Obsérvese que aquí aparece, en *Quantity*, un ciclo de llegadas que debe definirse previamente.

Mediante el menú *Build-More Elements-Arrival Cycles* se define el *Ciclo_1* de llegadas, así:

Identification	Quantity %	Cumulative	Table
Ciclo_1	Quantity	No	Defined

Al hacer clic en *Table* se abre una ventana con dos columnas: *Time (Hours)* y *Quantity (%)*. En estas dos columnas se coloca la información del número de clientes que llegan al sistema a la hora allí indicada. Se asume como hora 1 de las 6 a las 7 de la mañana.

Tiempo	1	2	3	4	5	6	7	8	9	10	11	12
Q %	40	40	40	20	20	20	20	20	20	40	40	40
Tiempo	13	14	15	16	17	18	19	20	21	22	23	24
Q %	20	20	20	10	10	10	10	10	10	10	10	10

Al hacer clic en *Logic* se abre una ventana. Mediante el martillito, dar ingreso a la siguiente información:

```
(*)     WRITELINE demanda, CALHOUR( )
        IF id=1 THEN
        begin
          graphic 2
          id = 0
        end
          else
        begin
          graphic 1
          id =1
        end
```

La instrucción WRITELINE escribe información de la hora de simulación a un archivo general que se identificó antes como *demanda* y comienza una nueva línea (Es como un WRITE + ENTER); de esta forma ProModel®, escribe la hora de entrada de un cliente al sistema, mediante la instrucción CALHOUR(). Con la instrucción *IF id=1 THEN* se chequea si la variable *id* tiene valor 1 o 0, y de acuerdo con el valor, se asigna la *gráfica 1* (hombre) o la *gráfica 2* (mujer). Esto permite un efecto visual de alternación de hombres y mujeres en la cola del cajero.

5. *BUILD (Variables)*

Para este modelo se consideran dos *variables*, definidas así:

Identification	Type	Initial Values	Statistics
Num_personas	Entera	0	Time Series
id	Entera	0	Time Series

En la variable *num_personas* se guarda el número clientes que llegan a la *cola_cajero* y que son atendidos por el *cajero*. La variable *id*, como se comentó antes es binaria y sirve para discriminar los clientes entre hombres y mujeres.

6. *BUILD (Processing)*

El proceso o *Processing* se define de la siguiente manera:

Entity	Location	Operation	Output	Destination	Rule	Move Log.
Cliente	Cola_cajero		Cliente	Cajero	First 1	Move for 0.5
Cliente	Cajero	**INC** num_personas **INT** min_ent **INT** min_sal min_ent = **CALMIN()** **WRITE** cliente_atendidos, **CALHOUR()** **WRITE** cliente_atendidos, min_ent **WAIT** 2 min_sal = **CALMIN()** **WRITE** cliente_atendidos, min_sal **WRITELINE** cliente_atendidos, min_sal-min_ent min_sal = **CALMIN()**	Cliente	Cola_salida	First 1	Move for 0.5
Cliente	Cola_salida		Cliente	EXIT	First 1	Move for 0.5

La instrucciòn INC incrementa la variable *num_personas* de clientes atendidos por el cajero. Por medio de INT *min_ent*, se crea la variable local entera *min_ent*, que va a guardar el tiempo de entrada del cliente al *cajero*, lo mismo hace INT *min_sal* para el tiempo de salida del cliente del *cajero*. Las variables locales se definen dentro del proceso, como se hace aquí, mientras que las variables globales se definen explícitamente.

La instrucción *min_ent* = CALMIN() coloca el tiempo de simulación en la variable local entera *min_ent*. Igual sucede con la instrucción *min_sal*= CALMIN(). La instrucción WAIT 2, hace esperar dos minutos al cliente en el cajero (aquí se asume tiempo de servicio *discreto*).

Mediante la instrucción WRITE, se escribe información en el archivo general llamado *cliente_atendidos*, previamente definido. El próximo ítem se escribirá inmediatamente después del último. En dicho archivo, el programa escribe el tiempo de salida de los clientes, guardado en la variable interna "min_sal". De manera similar, mediante WRITELINE se escribe la diferencia entre el tiempo de salida y el tiempo de entrada, o sea el tiempo de servicio, que aparece como *tiempo ocupado* en el archivo *C:\Temp\ clientes_ atendidos.txt*. El *Move for 0.5* incrementa en medio minuto el paso de la entidad *cliente* a través del sistema y sólo tiene efecto visual durante la simulación.

7. *BUILD (General Information)*

Para definir la *lógica de iniciación* incluir la siguiente información:

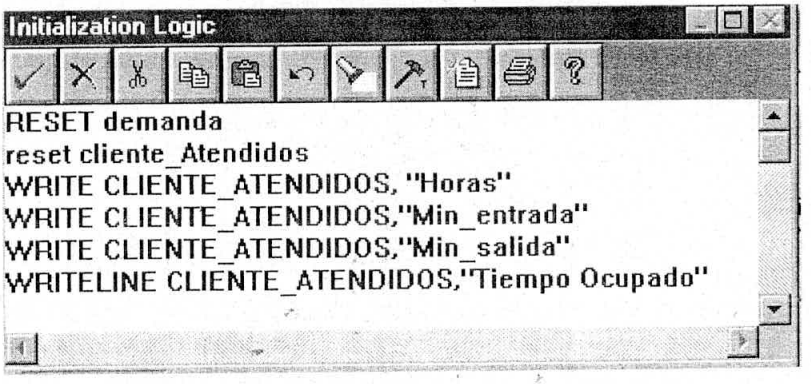


Figura 3.16 Lógica de iniciación del programa.

Las instrucciones RESET *demandá* y RESET *cliente_atendidos*, borran el contenido existente del correspondiente archivo EXCEL. Las instrucciones WRITE *cliente_atendidos*, "Horas", WRITE *cliente_atendidos*, *Min_entrada* y WRITE *cliente_atendidos*, "*Min_salida*" hacen que el programa escriba en el archivo *cliente_atendidos* de Excel, en el mismo renglón: *Horas Min_entrada Min_salida.*

La instrucción WRITELINE *cliente_atendidos*, *tiempo ocupado* escribe el tiempo ocupado de cada cliente y pasa a la siguiente línea. Para salir, de esta ventana se hace clic en el chulo de color verde que aparece allí.

8. *SIMULATION (Options)*

Definir el *tiempo de corrida* en 24 horas, señalar *Calendar Date* y dar *Run.*

Archivo "*clientes_atendidos*" para tiempos discretos

El archivo de escritura *cliente_atendidos* que creó el programa, se abre como una tabla en Excel. A partir de esta información se obtienen los gráficos, en Excel, que aparecen en las figuras 3.17 y 3.18.

"HORAS", "	Min entrada", "Min		salida",	"Tiempo ocupado",
0.00	2.00	0	4.00	2.00
0.00	5.00	0	7.00	2.00
0.00	7.00	0	9.00	2.00
0.00	10.00	0	12.00	2.00
0.00	12.00	0	14.00	2.00
0.00	15.00	0	17.00	2.00
0.00	17.00	0	19.00	2.00
0.00	20.00	0	22.00	2.00
0.00	22.00	0	24.00.00	2.00
0.00	25.00.00	0	27.00.00	2.00
0.00	27.00.00	0	29.00.00	2.00
0.00	30.00.00	0	32.00.00	2.00
0.00	32.00.00	0	34.00.00	2.00
0.00	35.00.00	0	37.00.00	2.00
0.00	37.00.00	0	39.00.00	2.00
0.00	40.00.00	0	42.00.00	2.00
0.00	42.00.00	0	44.00.00	2.00
0.00	45.00.00	0	47.00.00	2.00
0.00	47.00.00	0	49.00.00	2.00

Figura 3.17 Paso del archivo de texto a Excel.

Contar de "Tipo Cliente"	Horas						
"Tipo Cliente"	4	5	6	7	8	9	Total general
"Cheque"	18	21	16	15	18	22	110
"Consignacion"	46	34	29	31	41	35	216
Total general	64	55	45	46	59	57	326

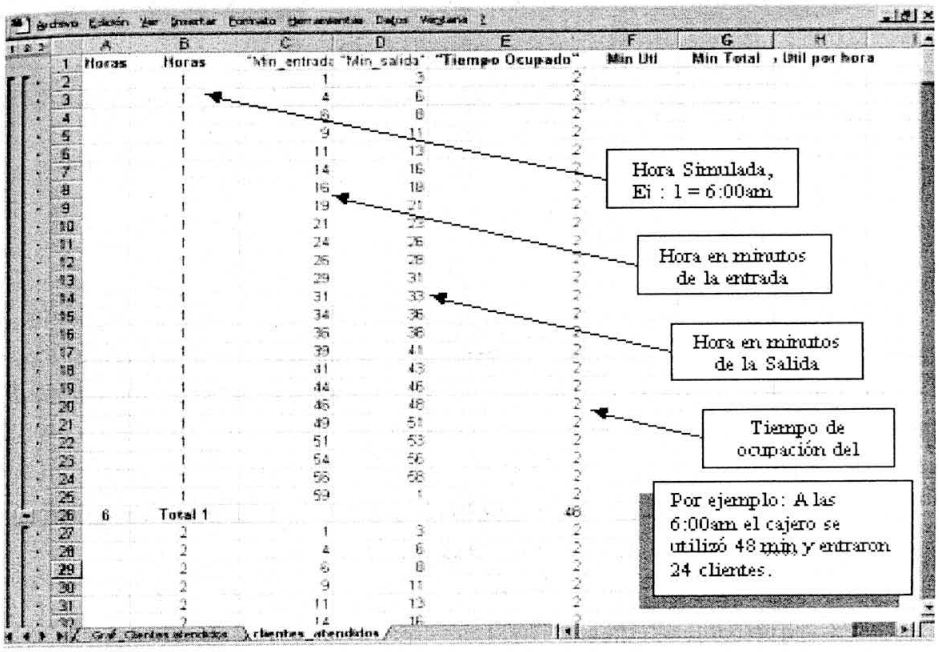


Tabla 3.1 Datos de la simulación con tiempos discretos. Archivo Excel.

Como se puede observar, las ayudas gráficas del ProModel®, se pueden potenciar bastante aprovechando las ventajas gráficas del Excel.

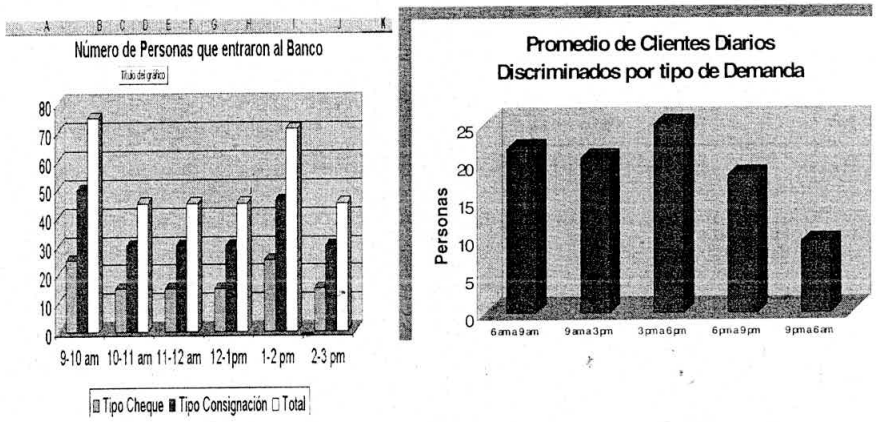


Figura 3.18 Diagrama de barras Excel de la demanda diaria con tiempos discretos.

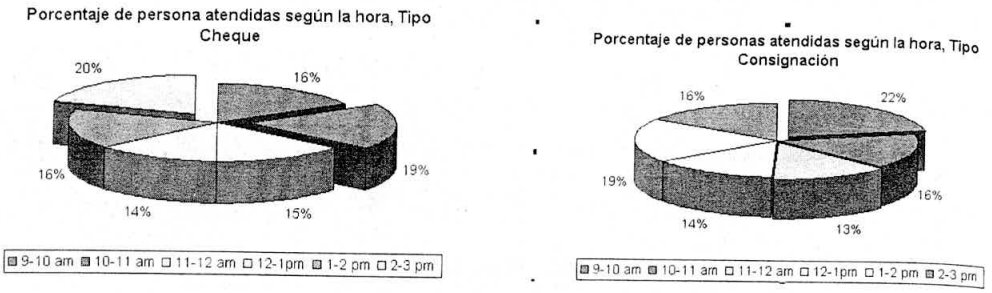


Figura 3.19 Diagrama de pastel Excel del porcentaje de personas atendidas según la hora.

La siguiente gráfica se obtiene al final de la corrida de ProModel®, respondiendo *Sí* a la pregunta ¿Se quieren estadísticas?. Seleccionar el icono de tendencias, marcar *Cola_cajero contents*, pasar a la ventana *Selected Values* con la flechita de la mitad y dando *OK*.

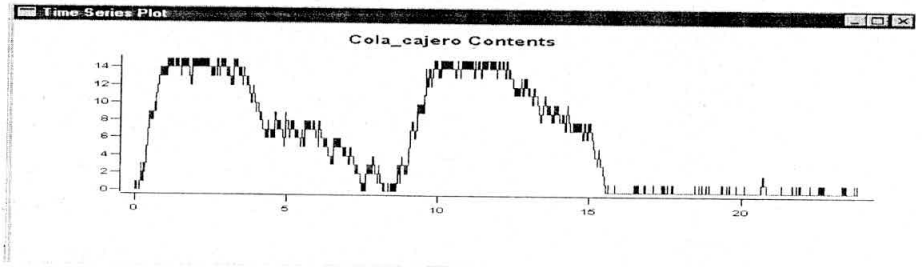


Figura 3.20 Gráfica de ProModel que muestra el número de personas en cola en el cajero.

LOCATION STATES BY PERCENTAGE (Single Capacity/Tanks)

Location Name	Scheduled Hours	% Operation	% Setup	% Idle	% Waiting	% Blocked	% Down
Cajero	23.88	64.27	0.00	35.73	0.00	0.00	0.00

FAILED ARRIVALS

Entity Name	Location Name	Total Failed
Cliente tipoA	Cola cajero	53

Figura 3.21 Datos estadísticos de ProModel que muestran los clientes que abandonaron el sistema, debido a que la cola era muy larga.

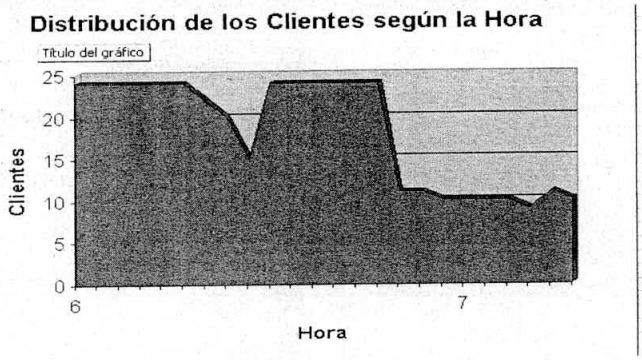


Figura 3.22 Gráfica Excel de la distribución de la llegada de clientes según la hora.

Archivo clientes_atendidos para tiempos aleatorios

El periodo de simulación de 24 horas, se definió mediante el menú *Simulation – Options – Run Lenght – Weekly Time*, comenzando la simulación un miércoles a las 6 a.m. y terminándola el siguiente jueves a las 6 a.m., con un tiempo de calentamiento *(Warm Up)* de 12 horas definido desde el martes a mediodía.

Mediante el menú *Simulation – Run*, ProModel® comienza la simulación animada. Al final aparece una ventana en la que se pregunta si se desean ver los resultados; al responder el usuario que sí, el programa abre una página de resultados, con una barra de herramientas en la parte superior.

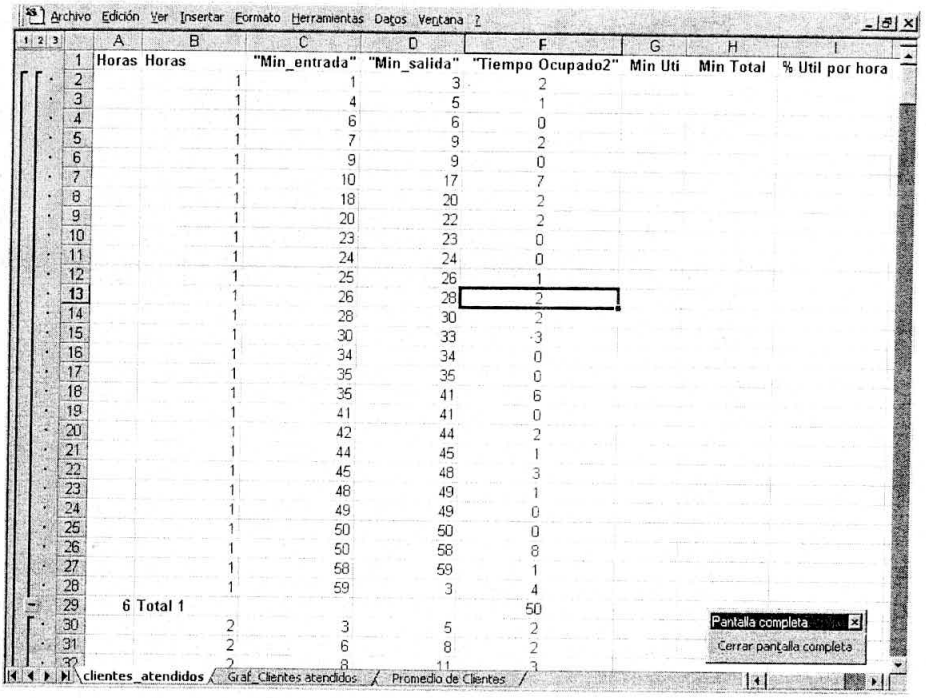
En este archivo de Excel, el programa ProModel® ha escrito los resultados de la simulación con tiempos aleatorios. Obsérvese su variación con relación a los resultados de tiempos discretos. Ver tablas 3.1 y 3.2.

Análisis de los resultados

a) Resultados con tiempos discretos

La simulación se corrió con los siguientes datos de entrada: alta demanda 40, mediana demanda 20, baja demanda 10, tiempo de servicio 2, capacidad de la cola de entrada 15, tiempo de simulación 24 horas. Los archivos Excel se abren en *C:\Temp* como archivos de texto.

En la figura 3.18 se observa la demanda diaria y el número máximo de clientes. El máximo número de clientes fue 23 en los periodos de 6 a 11 a.m y de 2 a 8 p.m. En la figura 3.19 se muestra el porcentaje de ocupación



	A	B	C	D	F	G	H	I
1	Horas	Horas	"Min_entrada"	"Min_salida"	"Tiempo Ocupado2"	Min Uti	Min Total	% Util por hora
2		1	1	3	2			
3		1	4	5	1			
4		1	6	6	0			
5		1	7	9	2			
6		1	9	9	0			
7		1	10	17	7			
8		1	18	20	2			
9		1	20	22	2			
10		1	23	23	0			
11		1	24	24	0			
12		1	25	26	1			
13		1	26	28	2			
14		1	28	30	2			
15		1	30	33	3			
16		1	34	34	0			
17		1	35	35	0			
18		1	35	41	6			
19		1	41	41	0			
20		1	42	44	2			
21		1	44	45	1			
22		1	45	48	3			
23		1	48	49	1			
24		1	49	49	0			
25		1	50	50	0			
26		1	50	58	8			
27		1	58	59	1			
28		1	59	3	4			
29	6 Total 1				50			
30		2	3	5	2			
31		2	6	8	2			
32		2	8	11	3			

Tabla 3.2 Datos de la simulación con tiempos aleatorios. Archivo Excel.

del cajero, con un máximo del 80% de 6 a 9 a.m y de 3 a 6 p.m., y en la figura 3.21, el número de clientes que abandonaron el sistema, en total 53.

En las estadísticas que arroja el ProModel®, se atendieron 437 clientes y 53 abandonaron el sistema porque encontraron 15 personas en cola.

b) Resultados con tiempos aleatorios

En la figura 3.23 se observan picos de demanda de 6 a 9 a.m. y de 3 a 6 p.m., con un máximo de 27 clientes. La figura 3.26 muestra una máxima ocupación del cajero de un 81% de 6 a 9 a.m. y de 3 a 6 p.m. La figura 3.24 muestra que 104 clientes abandonaron el sistema.

Archivo Excel.

) a 6 p.m., y en la
el istema, en to-

ieron 437 clientes
ic ıs en cola.

9 a.m. y de 3 a 6
:s ı una máxima
m. La figura 3.24

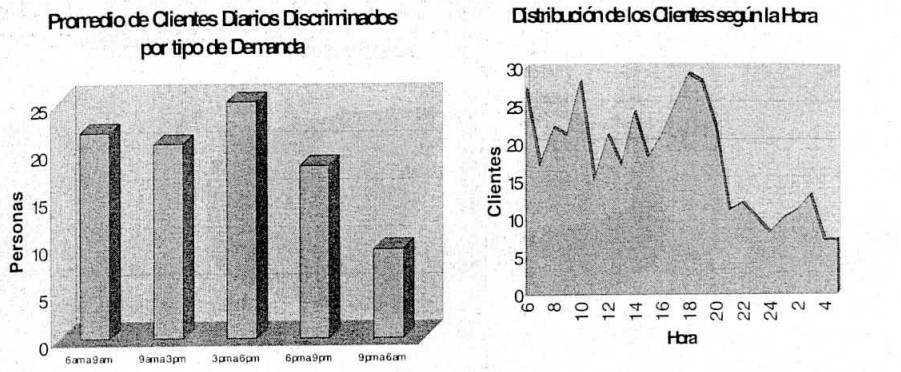


Figura 3.23 Diagrama de barras Excel de la demanda diaria con tiempos aleatorios.

FAILED ARRIVALS

Entity Name	Location Name	Total Failed
Cliente	Cola cajero	104

Figura 3.24 Datos estadísticos de ProModel que muestran el número de clientes que abandonaron el sistema porque la cola era muy larga, con tiempos aleatorios.

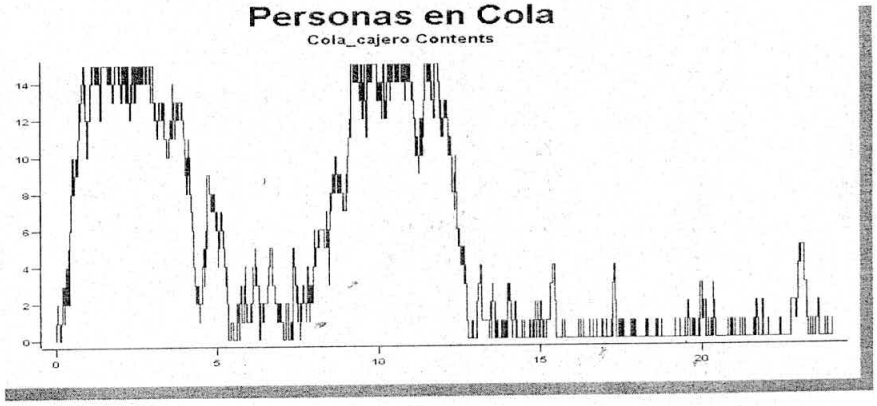


Figura 3.25 Número de personas en cola, en el modelo aleatorio.

Caso 3. Cajeros de un banco

Colas M/M/n

José Rodríguez, jefe del departamento de operaciones del Banco Urbano, desea calcular el número de cajeros que se deben colocar en la sede principal con el fin de prestar un óptimo servicio a sus clientes durante las horas de servicio.

a) Tipos de llegadas de clientes

Se distinguen dos tipos de llegadas de clientes, que dependen de la hora del día:

1. **Alta demanda** (9 a.m. a 10 a.m. y de 1 p.m. a 2 p.m.) Este horario corresponde a la primera hora de servicio, cuando se abre el banco y las personas que han hecho cola mientras abren las puertas, ingresan; la otra hora al medio día cuando las personas tienen su pausa para el almuerzo y la aprovechan para hacer sus diligencias bancarias. En este horario, 75 clientes por hora, en promedio, utilizan el banco, 50 consignan y 25 cobran cheques.
2. **Mediana demanda** (10 a.m. a 1 p.m. y de 2 p.m. a 3 p.m.) En este horario llegan 45 clientes por hora, en promedio; 30 a consignar y 15 a cobrar cheques.

b) Tiempo de servicio y costos

El tiempo promedio de servicio de cada cajero se ha estimado en dos minutos por cliente para consignaciones y en cinco minutos para cambio de cheques.

El costo fijo de funcionamiento de cada cajero se estima en $ 50.000/hora.

El costo de oportunidad por cada cliente que abandona el cajero sin ser atendido se estima en $ 20.000 por cliente. Un cliente abandona el banco cuando encuentra mas de quince clientes en la cola.

c) Información requerida

José quiere conocer:

- Número promedio de clientes atendidos diariamente, discriminados por tipo de demanda, para cada una de las situaciones que se describen más adelante.
- Número de cajeros necesarios para prestar un óptimo servicio, discriminados por tipo de demanda.
- Número de clientes diarios que no fueron atendidos por el banco y su costo de oportunidad, para cada una de las situaciones que se describen más adelante.

d) Situaciones

Tiempos discretos.

	Alta demanda	**Mediana demanda**
No. de llegadas por hora	50 a consignar 25 por cheques	30 a consignar 15 por cheques
Tiempo de servicio (min.)	Consignaciones 2 Cheques 5	Consignaciones 2 Cheques 5

Tiempos aleatorios (Cola M/M/n).

	Alta demanda	**Mediana demanda**
No. de llegadas por hora	Poisson (50) consignar* Poisson (25) cheques	Poisson (30) consignar Poisson (15) cheques
Tiempo de servicio (min.)	Consignaciones **e(2)**** Cheques **e(5)**	Consignaciones **e(2)** Cheques **e(5)**

* Distribución de Poisson con media 50.

** e(2)=Distribución exponencial con media dos.

En qué consiste una cola M/M/n

Una cola **M/M/***n* es un sistema conformado por *n* servidores en paralelo, en este caso los cajeros, y una serie de clientes que llegan de manera aleatoria a una cola ubicada al frente de los cajeros. El cliente que está a la cabeza de la fila pasa al primer cajero que quede desocupado, a solicitar su servicio. Siempre se están atendiendo *n* clientes.

Los clientes llegan al sistema de uno en uno. El número de llegadas de clientes, que se suceden en un intervalo de tiempo *t*, es una variable aleatoria independiente y discreta que se modela muy bien con una distribución de Poisson, con una tasa entre llegadas Lambda (*L*) y número de llegadas tL[5]. Como las llegadas dependen del número de clientes en el sistema *m*, la tasa entre llegadas se acostumbra llamar *Lm*.

Los *n* servidores atienden a los clientes con una duración de servicio, otro conjunto de variables aleatorias independientes y continuas que se modelan con una distribución exponencial negativa con tasa de servicio *Mu* (*M*). Como el sistema tiene *n* servidores en paralelo, suponiendo que haya *m* clientes en el sistema, esto es, en espera y en servicio, la tasa de salidas del sistema es *Mm*. Tan pronto un cliente es servido, abandona el sistema.

Modelando el Banco con Promodel®

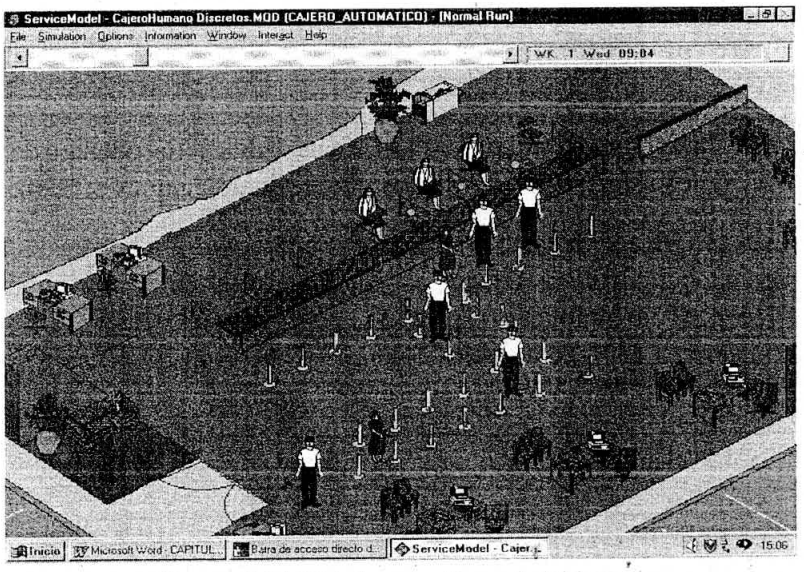


Figura 3.27 Distribución (*layout*) del banco.

5. Wayne L. Winston, *Investigación de operaciones, aplicaciones y algoritmos*, México, Grupo Editorial Iberoamérica, 1994, p. 1114.

1. *BUILD (Locations)*

Se consideran siete instalaciones: *cola_entrada, cajero 1, cajero 2, cajero 3, cajero 4, cola_salida* y *escritorio de cajeros.* Los datos iniciales son:

Name	Capacity	Units	Down Times	Statistics
Cola_entrada	15	1	None	Time series
Cajero 1	1	1	None	Time series
Cajero 2	1	1	None	Time series
Cajero 3	1	1	None	Time series
Cajero 4	1	1	None	Time series
Cola_salida-banco	Infinite	1	None	Time series
Escritorio_cajeros	1	1	None	Time series

2. *BUILD (Entities)*

Se trabaja con dos entidades: *cliente_consignación* y *cliente_cheque.*

3. *BUILD (Atributtes)*

En este modelo se considera un atributo *tipo_cliente*, definido así:

Identification	Type	Classification	Notes
Tipo_cliente	Integer	Entity	1= consignación 2= cheque

Este atributo sirve para "marcar" a las entidades; a los clientes de consignación se les asigna *uno (1)* y a los clientes con cheques se les asigna *dos (2).*

4. *BUILD (Arrivals)*

Las llegadas se definen así:

Entity	Location	Q. Each	F. T.	Oc.	Freq.	Logic	Dis.
Cliente_Consignación	Cola_entrada	1, ciclo_cajero_consignación	0	1	24 hr	Tipo_cliente= 1	No
Cliente_cheque	Cola_entrada	1, ciclo_cajero_cheque	0	1	24 hr	Tipo_cliente = 2	No

En el momento de definir la cantidad de llegadas simultáneas de la entidad *cliente_consignación*, se informa que hay un *ciclo de llegadas* o *arrivals cycle*, llamado *ciclo_cajero_consignación*, que es necesario especificar mediante el menú *Build-More Elements-Arrival Cycle*. Lo mismo sucede a las llegadas de la entidad *cliente_cheque*.

5. *BUILD (More elements-Arrival cycle)*

Los ciclos de llegada se especifican de la siguiente manera:

Identification	Qt%	Cumulative	Table
Ciclo_cajero_consignación	Quantity	No	Defined
Ciclo_cajero_cheque	Quantity	No	Defined

Nuevamente, es necesario definir la *tabla de llegadas* (haciendo doble clic en *Table*) para cada uno de los ciclos.

Ciclo_cajero_consignación		Ciclo_cajero_cheque	
Time (Hours)	Qt(%)	Time (Hours)	Qt(%)
1		1	
2		2	
3		3	
4	50	4	25
5	30	5	15
6	30	6	15
7	30	7	15
8	50	8	25
9	30	9	15

Al observar la forma como se definieron estos ciclos de llegada, se ve claramente que el número de clientes de cheques es la mitad del número de clientes de consignaciones y que la hora 4 corresponde de 9 a.m a 10 a.m. En la lógica de las llegadas, se advierte que los clientes de consignación son de *tipo 1* y que los clientes de cheque son de *tipo 2*.

6. *BUILD (Variables)*

Por otra parte, se tuvieron en cuenta dos variables enteras con valor inicial 1:

Icon	Identification	Type	Initial Value	Statistics
No	Zoom_off1	Integer	1	
No	Zoom_off2	integer	1	

Zoom_off1 y *zoom_off2* se definen para ayudar a mostrar diferentes vistas de la distribución o *layout*.

7. *VIEW (Zoom)*

Por medio de este menú se especifica el área o región que se quiere mostrar, utilizando las flechitas de la ventana y fijando el *zoom* en *Custom Zoom;* luego, mediante *View-Views-Define-Add-Set View* se da el nombre a la *vista*. En el modelo se definieron tres *vistas*: *todo*, *cajeros*, *cuatro cajeros*.

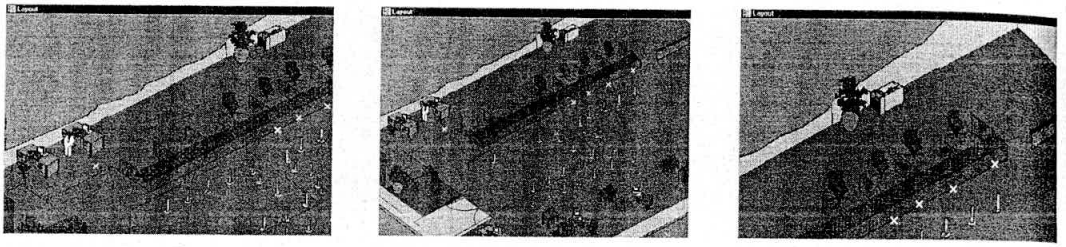


Figura 3.28 Vistas: *todo*, *cajeros*, *cuatro cajeros*.

En la primera vista se muestra un panorama general del banco; en la segunda se hace un acercamiento al escritorio de cajeros y a la cola de entrada; en la tercera, como el nombre lo indica, el acercamiento se hace a los cuatro cajeros.

8. *BUILD (Path networks)*

Se define también la ruta *Net 1* por donde se movilizan los cajeros, con los siguientes datos:

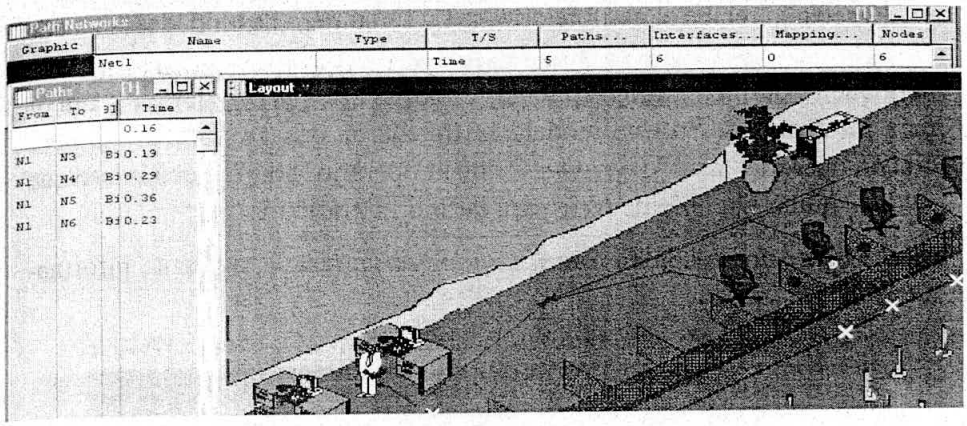
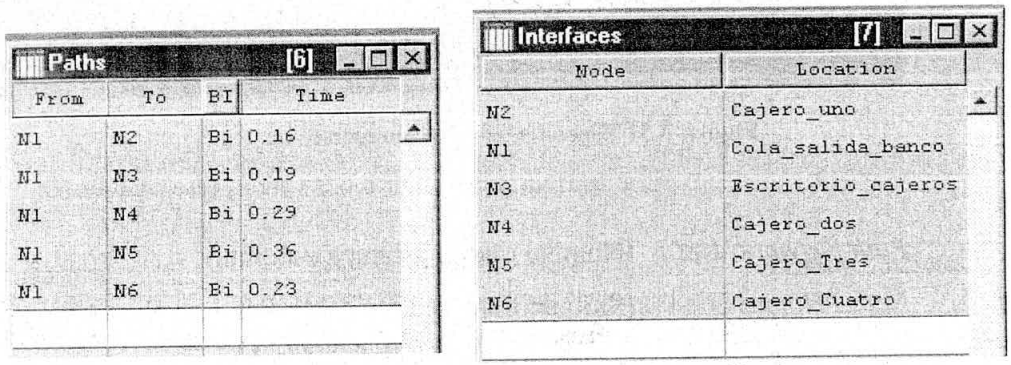


Figura 3.29 Ruta de movimiento de los cajeros.

Graphic	Name	Type	T/S	Paths	Interfaces	Mapping	Nodes
	Net 1	Passing	Time	5	6	0	6

Al hacer doble clic sobre *Paths* y sobre *Interfaces*, se abren las siguientes ventanas:



From	To	BI	Time
N1	N2	Bi	0.16
N1	N3	Bi	0.19
N1	N4	Bi	0.29
N1	N5	Bi	0.36
N1	N6	Bi	0.23

Node	Location
N2	Cajero_uno
N1	Cola_salida_banco
N3	Escritorio_cajeros
N4	Cajero_dos
N5	Cajero_Tres
N6	Cajero_Cuatro

Figura 3.30 Trayectorias e interfaces de la ruta *Net 1*.

9. *BUILD (Resources)*

En el modelo también se utiliza un recurso que se define de la siguiente manera:

Icon	Name	Units	Down Times	Statistics	Specifications	Search	Logic	Points	Notes
Yes	*Res1*	3	None	By unit	Net1, N3	None	5	2	

El recurso recibe el nombre de *Res 1* y su número de unidades se puede variar cada vez. En este caso se consideran tres *cajeros*. De la misma manera que se definen gráficos para las entidades, se pueden definir gráficos para los recursos. En este modelo se trabajaron dos *gráficos*: *Graphic 1* con un operario "azul" y *graphic 2* con el operario "rojo". Recuérdese que se debe quitar el chulo de *New* para definir el gráfico 2.

Al hacer doble clic en *Specifications* se ingresa la siguiente información:

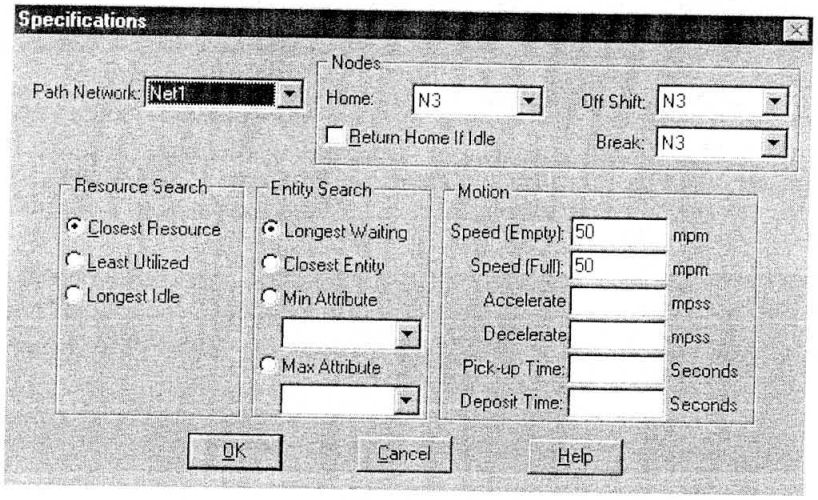
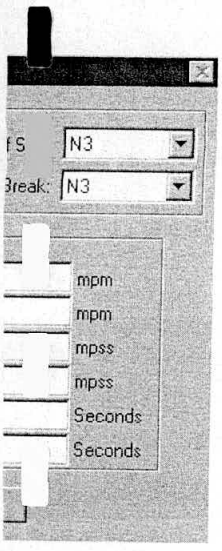


Figura 3.31 Especificaciones del recurso *Res 1*.

Path Network: *Net 1*	(Nombre que se asigna a esta ruta)
Nodes Home N3	(Sitio de trabajo del recurso *Res 1* es el *escritorio de cajeros*, o sea, el nodo *N3*)
Off Shift N3	(Si el cajero no está en turno, regresa al *escritorio de cajeros, N3)*
Break N3	(Si el cajero está en pausa, regresa al *escritorio de cajeros, N3*)

o unidades se puede
eros. De la misma ma-
pueden definir gráficos
os *ráficos*: *Graphic 1*
'rojo". Recuérdese que
:o ?.

l.. siguiente informa-



R 1.

st uta)

o *Res 1* es el *escrito-*
lc *13*)

io, regresa al *escrito-*

regresa al *escritorio*

Al hacer doble clic en *Logics* se introduce la siguiente información:

Node	Entry Logic	Exit Logic
N2	Graphic 1	Graphic 2
N3	Graphic 1	Graphic 2
N4	Graphic 1	Graphic 2
N5	Graphic 1	Graphic 2
N6	Graphic 1	Graphic 2

El efecto de esta instrucción en el modelo es el de cambiar el gráfico del recurso *Res 1*; por ejemplo, en el *nodo 2*, o sea el *cajero_uno*, el recurso entra "azul" y sale "rojo".

En este Modelo también se definieron las coordenadas de dos puntos especiales: La cabeza de la *cola_ entrada N1* y el *escritorio de cajeros N3*. Para observar la asignación de las coordenadas se hace doble clic en *Points* y automáticamente aparece la información de las coordenadas de los nodos. Si los valores son diferentes en su gráfico, no se preocupe.

Node	Point
N1	0,0
N3	-62,-31

10. *BUILD (More elements-External files)*

Mediante este menú se definen los siguientes archivos:

Identification	Type	File Name	Prompt	Notes
Archivo_caso2	General Write	C:\Temp\ CasoDos.txt		
Archivo_caso2_personas_cola	General Write	C:\Temp\ CasoDosCola.txt		

11. *BUILD (Processing)*

El proceso se define de la siguiente manera:

Entity	Location	Operation	Output	Destination	Rule	Move Logic

All	Cola_entrada	**IF** Zoom_Off 1=1 **AND** Zoom_Off2 = 1 **THEN**
		BEGIN
		VIEW *"Cajeros"*
		Zoom_Off 1= 0
		END
		WRITE Archivo_caso2_personas_cola, **CALHOUR()**
		IF Tipo_cliente = 1 **THEN**
		WRITELINE Archivo_caso2_personas_cola, " Consignación"
		ELSE
		WRITELINE Archivo_caso2_personas_cola, " Cheque"
			All
			All
			All
			All
				Cajero_uno
				Cajero_dos
				Cajero_tres
				Cajero_cuatro
					Empty
					Empty
					Empty
					Empty
						Move For 0.5
						Move For 0.5
						Move For 0.5
						Move For 0.5

All	Cajero_uno	**WRITE** Archivo_caso2, **CALHOUR()**
		IF Zoom_Off2 =1 **THEN**
		BEGIN
		VIEW *"Cuatro cajeros"*
		Zoom_Off 2=0
		END
		IF Tipo_cliente = 1 **THEN**
		(**WRITELINE** Archivo_caso2, "Consignación"
		USE ***Res1*** for 2)
		ELSE
		(**WRITELINE** Archivo_caso2, "Cheque"
		USE ***Res1*** for 5)
			All	Cola_salida_banco	First 1	Move For 0.5

All	Cajero_dos	**WRITE** Archivo_caso2, **CALHOUR()**
		IF Tipo_cliente = 1 **THEN**
		(**WRITELINE** Archivo_caso2, "Consignación"
		USE ***Res1*** for 2)
		ELSE
		(**WRITELINE** Archivo_caso2, "Cheque"
		USE ***Res1*** for 5)
			All	Cola_salida_banco	First 1	Move For 0,5

All	Cajero_tres	**WRITE** Archivo_caso2, **CALHOUR()**
		IF Tipo_cliente = 1 **THEN**
		(**WRITELINE** Archivo_caso2, "Consignación"
		USE ***Res1*** for 2)
		ELSE
		(**WRITELINE** Archivo_caso2, "Cheque"
		USE ***Res1*** for 5)
			All	Cola_salida_banco	First 1	Move For 0,5

All	Cajero_cuatro	**WRITE** Archivo_caso2, **CALHOUR()**
		IF Tipo_cliente = 1 **THEN**
		(**WRITELINE** Archivo_caso2, "Consignación"
		USE ***Res1*** for 2)
		ELSE
		(**WRITELINE** Archivo_caso2, "Cheque"
		USE ***Res1*** for 5)
			All	Cola_salida_banco	First 1	Move For 0,5

All	Cola_salida_banco		All	EXIT	First 1	Move For 0,5

D	ination	Rule	Move Logic
Cajero_uno Cajero_dos Ca _tres Ca _cuat ro		Empty Empty Empty Empty	Move For 0.5 Move For 0.5 Move For 0.5 Move For 0.5
ola ban	lida	First 1	Move For 0.5
ıce	da	First 1	Move For 0,5
:o	a	First 1	Move For 0,5
al		First 1	Move For 0,5
		First 1	Move For 0,5

En el proceso de este modelo hay algunos aspectos importantes: en el campo *Entity*, se colocó *All*. Esta instrucción permite que estén ingresando al sistema, "aleatoriamente"[6], las dos entidades definidas *cliente_consignación* y *cliente_cheque*. Generalmente, las entidades entran al sistema en el mismo orden en que se definen en los *arrivals*.

La primera operación en *cola_entrada*, enfoca la *vista* en *cajeros* y chequea el tipo de cliente que está entrando al sistema. Si el tipo de cliente es 1, éste se escribe como *Consignación* en el *Archivo_caso2_personas_cola*, de lo contrario, se escribe *cheque*. En este mismo archivo se escribe el tiempo de llegada a la cola.

La segunda operación en *cajero-uno*, enfoca la *vista* en *cuatro cajeros* y se utiliza el recurso "res 1" durante 2 minutos, esto se programa mediante la instrucción *USE Res1 for 2*. Como en la operación anterior, si el tipo de cliente es 1, éste se escribe como *Consignación*, en el *Archivo_caso2*, de lo contrario, se escribe *cheque*, en este mismo archivo se escribe el tiempo de llegada a la cola. Para los demás cajeros, las operaciones son similares.

12. *BUILD (Cost)*

En el modelo también se definió un costo de $ 50.000 por hora por *cajero* o recurso *Res 1*, utilizando la siguiente información:

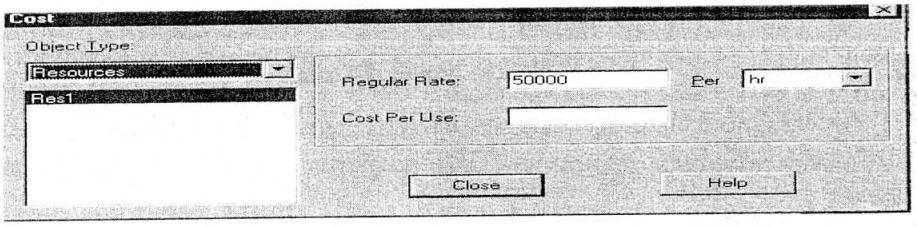


Figura 3.32 Información de costos del recurso *Res 1*.

13. *BUILD (General Information-Initialization Logic)*

Mediante éste menú, se actualizan los archivos externos con las siguientes instrucciones:

6 Nota: se ha colocado aleatoriamente entre comillas porque se sabe, por la información de los ciclos de llegada, que están llegando al sistema el doble de consignaciones que de cheques.

RESET Archivo_caso2 (Borra la información anterior)
WRITE Archivo_caso2, "Horas" (Escribe *Horas* en el archivo)
WRITELINE Archivo_caso2, "Tipo Cliente"
RESET Archivo_caso2_personas_cola
WRITE Archivo_caso2_personas_cola, "Horas"
WRITELINE Archivo_caso2_personas_cola, "Tipo Cliente"

14. *SIMULATION (Options)*

Con éste menú se define la *duración de la simulación*. Primero se define con *Browse* el nombre del archivo en el cual se almacenan los resultados de la simulación, en este caso, C:\Temp*Cajeros.mod*. En *Warm up start* se puede colocar el tiempo de "calentamiento" del sistema, antes de iniciar la simulación. Se debe señalar *Calendar Date,* para que funcione la instrucción *CALHOUR()*. Por medio de *Sim. Begin* y *Sim. End* se coloca el tiempo de la simulación, que para este modelo fue 24 horas.

Análisis de los resultados

El modelo se corrió variando el número de cajeros desde uno hasta cinco, con tiempos de llegada y tiempos de servicio primero discretos y luego aleatorios. Se obtuvieron los siguientes resultados:

Resultados con tiempos discretos

En primer lugar se utilizaron tiempos discretos, tanto en la llegada de clientes como en los tiempos de servicio de los cajeros, variando el número de cajeros.

a) Análisis de clientes atendidos

DOS CAJEROS

ENTITY ACTIVITY

Entity Name	Total Exits	Current Quantity In System	Average Minutes In System
Cliente	0	0	-
Cliente Consignacion	136	0	27.193316
Cliente Cheque	77	0	30.263623
Fuera de Servicio	0	0	-

TRES CAJEROS

ENTITY ACTIVITY

Entity Name	Total Exits	Current Quantity In System	Average Minutes In System
Cliente	0	0	-
Cliente Consignacion	204	0	13.104657
Cliente Cheque	104	0	16.560385
Fuera de Servicio	0	0	-

CUATRO CAJEROS

Entity Name	Total Exits	Current Quantity In System	Average Minutes In System
Cliente	0	0	-
Cliente Consignacion	216	0	7.460046
Cliente Cheque	110	0	10.391727
Fuera de Servicio	0	0	-

CINCO CAJEROS

ENTITY ACTIVITY

Entity Name	Total Exits	Current Quantity In System	Average Minutes In System
Cliente	0	0	-
Cliente Consignacion	220	0	5.968768
Cliente Cheque	110	0	8.791882
Fuera de Servicio	0	0	-

Figura 3.33 Tiempos promedio en el sistema y número de clientes atendidos. Caso discreto.

b) Análisis de clientes no atendidos

DOS CAJEROS

FAILED ARRIVALS

Entity Name	Location Name	Total Failed
Cliente Consignacion	Cola Entrada	84
Cliente Cheque	Cola Entrada	33

TRES CAJEROS

FAILED ARRIVALS

Entity Name	Location Name	Total Failed
Cliente Consignacion	Cola Entrada	16
Cliente Cheque	Cola Entrada	6

CUATRO CAJEROS

FAILED ARRIVALS

Entity Name	Location Name	Total Failed
Cliente Consignacion	Cola Entrada	4
Cliente Cheque	Cola Entrada	0

CINCO CAJEROS

FAILED ARRIVALS

Entity Name	Location Name	Total Failed
Cliente Consignacion	Cola Entrada	0
Cliente Cheque	Cola Entrada	0

Figura 3.34 Clientes no atendidos. Tiempos de llegada y de servicio discretos.

c) Análisis del costo de operación de los cajeros

DOS CAJEROS

RESOURCES COSTING

Resource Name	Units	$ NonUse Cost	% NonUse Cost	$ Usage Cost	% Usage Cost	$ Total Cost	% Total Cost
Res1.2	1	199210.833333	50.32	272791.666667	49.77	472002.500000	50.00
Res1.2	1	196652.500000	49.68	275350.000000	50.23	472002.500000	50.00
Res1	2	395863.333333	100.00	548141.666667	100.00	944005.000000	100.00
SUM	-	395863.333333	100.00	548141.666667	100.00	944005.000000	100.00

TRES CAJEROS

RESOURCES COSTING

Resource Name	Units	$ NonUse Cost	% NonUse Cost	$ Usage Cost	% Usage Cost	$ Total Cost	% Total Cost
Res1.1	1	203800.833333	33.39	257791.666667	33.29	461592.500000	33.33
Res1.2	1	201192.500000	32.96	260400.000000	33.63	461592.500000	33.33
Res1.3	1	205409.166667	33.65	256183.333333	33.08	461592.500000	33.33
Res1	3	610402.500000	100.00	774375.000000	100.00	1384777.500000	100.00
SUM	-	610402.500000	100.00	774375.000000	100.00	1384777.500000	100.00

CUATRO CAJEROS

RESOURCES COSTING

Resource Name	Units	$ NonUse Cost	% NonUse Cost	$ Usage Cost	% Usage Cost	$ Total Cost	% Total Cost
Res1.1	1	238534.166667	24.21	212791.666667	25.96	451325.833333	25.00
Res1.2	1	246759.166667	25.04	204566.666667	24.95	451325.833333	25.00
Res1.3	1	249200.833333	25.29	202125.000000	24.65	451325.833333	25.00
Res1.4	1	250975.833333	25.47	200350.000000	24.44	451325.833333	25.00
Res1	4	985470.000000	100.00	819833.333333	100.00	1805303.333333	100.00
SUM	-	985470.000000	100.00	819833.333333	100.00	1805303.333333	100.00

CINCO CAJEROS

RESOURCES COSTING

Resource Name	Units	$ NonUse Cost	% NonUse Cost	$ Usage Cost	% Usage Cost	$ Total Cost	% Total Cost
Res1.1	1	295186.666667	20.65	156125.000000	18.88	451311.666667	20.0(
Res1.2	1	270911.666667	18.95	180400.000000	21.82	451311.666667	20.0(
Res1.3	1	290853.333333	20.34	160458.333333	19.41	451311.666667	20.0(
Res1.4	1	274295.000000	19.19	177016.666667	21.41	451311.666667	20.0(
Res1.5	1	298461.666667	20.88	152850.000000	18.49	451311.666667	20.0(
Res1	5	1429708.333333	100.00	826850.000000	100.00	2256558.333333	100.0(
SUM	-	1429708.333333	100.00	826850.000000	100.00	2256558.333333	100.0(

Figura 3.35 Costos de operación del banco con tiempos de llegada y de servicio discretos.

Matriz de toma de decisiones

Num Cajeros	Min. En el Sistema	Cliente no atendidos	Costo no Uso	Costo Uso	Costo Cliente no Atendido	Costo Total
Dos	28,7	117	395.863	548.141	2.340.000	3.284.004
Tres	14,8	22	610.402	774.375	440.000	1.824.777
Cuatro	8,9	4	985.470	819.833	80.000	1.885.303
Cinco	7,3	0	1.429.708	826.850	0	2.256.558

Figura 3.36 Cuadro comparativo con tiempos de llegada y de servicio discretos.

Los datos que aparecen en las figuras 3.33, 3.34 y 3.35 corresponden a las estadísticas que arroja el ProModel®, cada vez que se *corre* la simulación, para el caso discreto.

En la figura 3.33 se observa que al aumentar el número de cajeros de cuatro a cinco, el número de clientes atendidos en cheques permanece igual (110), mientras que el número de clientes atendidos en consignaciones aumenta únicamente en cuatro, esto es, pasa de 216 a 220.

La figura 3.34 muestra que los datos de los clientes no atendidos son consistentes con los anteriores. Cuando trabajan cuatro cajeros se quedan solamente cuatro clientes de consignación sin atender, mientras que cuando trabajan cinco cajeros, no se queda ningún cliente sin atender.

En la figura 3.35 se resumen los costos de operación para todos los posibles números de cajeros.

En la figura 3.36 se hace la comparación de las distintas alternativas. Se escoge como óptima la política de emplear cuatro cajeros, asumiendo como costo de oportunidad el riesgo de no atender cuatro clientes de cheques.

Las figuras 3.37, 3.38 y 3.39 presentan los resultados de clientes atendidos y no atendidos, para el caso aleatorio.

En la figura 3.40 se incluye un cuadro resumen de las distintas alternativas para tiempos de llegada y de servicio aleatorios; como en el caso discreto, se ha seleccionado la mejor alternativa de trabajar con cuatro cajeros.

Resultados con tiempos aleatorios

En segundo lugar, se utilizaron tiempos aleatorios, tanto en la llegada de clientes como en los tiempos de servicio de los cajeros, y se varió el número de cajeros.

sto Cliente no Atendido	Costo Total
2.340.000	3.284.004
440.000	1.824.777
80.000	
0	2.256.558

le servicio discretos.

corresponden a
se *corre* la simula-

mero de cajeros de
e permanece igual
c signaciones au-
).

s o atendidos son
cajeros se quedan
mientras que cuan-
n tender.

ión para todos los

tintas alternativas.
cajeros, asumiendo
l ntes de cheques.

s de clientes aten-

distintas alternati-
en el caso discre-
atro cajeros.

o en la llegada de
varió el núme-

a) Análisis de clientes atendidos

DOS CAJEROS

ENTITY ACTIVITY

Entity Name	Total Exits	Current Quantity In System	Average Minutes In System
Cliente	0	0	-
Cliente Consignacion	138	0	28.462362
Cliente Cheque	69	0	31.339652
Fuera de Servicio	0	0	-

TRES CAJEROS

ENTITY ACTIVITY

Entity Name	Total Exits	Current Quantity In System	Average Minutes In System
Cliente	0	0	-
Cliente Consignacion	184	0	14.067272
Cliente Cheque	106	0	18.111075
Fuera de Servicio	0	0	-

CUATRO CAJEROS

ENTITY ACTIVITY

Entity Name	Total Exits	Current Quantity In System	Average Minutes In System
Cliente	0	0	-
Cliente Consignacion	210	0	8.401871
Cliente Cheque	117	0	11.146906
Fuera de Servicio	0	0	-

CINCO CAJEROS

ENTITY ACTIVITY

Entity Name	Total Exits	Current Quantity In System	Average Minutes In System
Cliente	0	0	-
Cliente Consignacion	210	0	6.552552
Cliente Cheque	116	0	8.224276
Fuera de Servicio	0	0	-

Figura 3.37 Tiempos promedios en el sistema, con tiempos de llegada y de servicio aleatorios.

b) Análisis de clientes no atendidos

DOS CAJEROS

FAILED ARRIVALS

Entity Name	Location Name	Total Failed
Cliente Consignacion	Cola Entrada	92
Cliente Cheque	Cola Entrada	61

TRES CAJEROS

FAILED ARRIVALS

Entity Name	Location Name	Total Failed
Cliente Consignacion	Cola Entrada	34
Cliente Cheque	Cola Entrada	18

CUATRO CAJEROS

FAILED ARRIVALS

Entity Name	Location Name	Total Failed
Cliente Consignacion	Cola Entrada	10
Cliente Cheque	Cola Entrada	1

CINCO CAJEROS

FAILED ARRIVALS

Entity Name	Location Name	Total Failed
Cliente Consignacion	Cola Entrada	5
Cliente Cheque	Cola Entrada	0

Figura 3.38 Clientes no atendidos, con tiempos de llegada y de servicio aleatorios.

c) Análisis según del costo de operación de los cajeros

DOS CAJEROS

RESOURCES COSTING

Resource Name	Units	$ NonUse Cost	% NonUse Cost	$ Usage Cost	% Usage Cost	$ Total Cost	% Total Cost
Res1.1	1	195371.666667	49.76	289410.000000	50.16	484781.666667	50.00
Res1.2	1	197268.333333	50.24	287513.333333	49.84	484781.666667	50.00
Res1	2	392640.000000	100.00	576923.333333	100.00	969563.333333	100.00
SUM	-	392640.000000	100.00	576923.333333	100.00	969563.333333	100.00

TRES CAJEROS

RESOURCES COSTING

Resource Name	Units	$ NonUse Cost	% NonUse Cost	$ Usage Cost	% Usage Cost	$ Total Cost	% Total Cost
Res1.1	1	208580.000000	34.08	247080.833333	32.73	455660.833333	33.33
Res1.2	1	198419.166667	32.42	257241.666667	34.08	455660.833333	33.33
Res1.3	1	205106.666667	33.51	250554.166667	33.19	455660.833333	33.33
Res1	3	612105.833333	100.00	754876.666667	100.00	1366982.500000	100.00
SUM	-	612105.833333	100.00	754876.666667	100.00	1366982.500000	100.00

CUATRO CAJEROS

RESOURCES COSTING

Resource Name	Units	$ NonUse Cost	% NonUse Cost	$ Usage Cost	% Usage Cost	$ Total Cost	% Total Cost
Res1.1	1	238646.666667	24.50	218792.500000	25.57	457439.166667	25.00
Res1.2	1	227335.833333	23.34	230103.333333	26.89	457439.166667	25.00
Res1.3	1	245940.833333	25.25	211498.333333	24.71	457439.166667	25.00
Res1.4	1	262017.500000	26.90	195421.666667	22.83	457439.166667	25.00
Res1	4	973940.833333	100.00	855815.833333	100.00	1829756.666667	100.00
SUM	-	973940.833333	100.00	855815.833333	100.00	1829756.666667	100.00

CINCO CAJEROS

RESOURCES COSTING

Resource Name	Units	$ NonUse Cost	% NonUse Cost	$ Usage Cost	% Usage Cost	$ Total Cost	% Total Cost
Res1.1	1	270449.166667	18.40	181409.166667	22.98	451858.333333	20.00
Res1.2	1	297800.000000	20.26	154058.333333	19.51	451858.333333	20.00
Res1.3	1	287194.166667	19.54	164664.166667	20.86	451858.333333	20.00
Res1.4	1	289482.500000	19.70	162375.833333	20.57	451858.333333	20.00
Res1.5	1	324835.000000	22.10	127023.333333	16.09	451858.333333	20.00
Res1	5	1469760.833333	100.00	789530.833333	100.00	2259291.666667	100.00
SUM	-	1469760.833333	100.00	789530.833333	100.00	2259291.666667	100.00

Figura 3.39 Costos de operación con tiempos de llegada y tiempos de servicio aleatorios.

Matriz de toma de decisión. Sistema aleatorio

Num Cajeros	Min. En el Sistema	Cliente no atendidos	Costo no Uso	Costo Uso	Costo Cliente no Atendido	Costo Total
Dos	29,85	153	392.640	576.923	3.060.000	4.029.563
Tres	16,05	52	612.105	754.876	1.040.000	2.406.981
Cuatro	9,77	11	973.940	855.815	220.000	2.049.755
Cinco	7,35	5	1.469.760	789.530	100.000	2.359.290

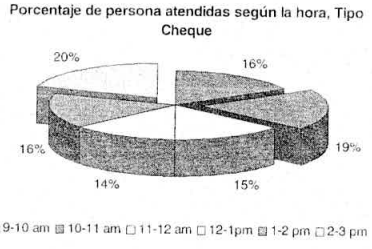
Figura 3.40 Cuadro comparativo con tiempos de llegada y de servicio aleatorios.

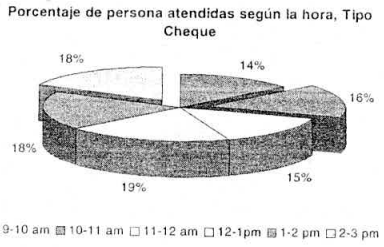
Las figuras 3.41 y 3.42 se obtuvieron a partir de los archivos *Excel*, donde ProModel® ha escrito la hora de llegada y los tipos de clientes, así como el tiempo que se demoró el sistema en atenderlos. Aquí se está haciendo uso de la poderosa herramienta gráfica del *Excel*, para facilitar el análisis de la información.

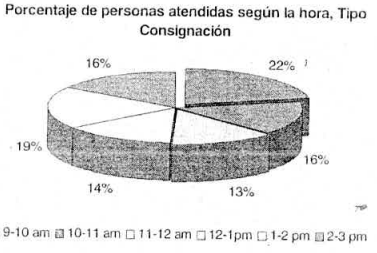
Gráficas con tiempos discretos

Contar de "Tipo Cliente"	Horas						
"Tipo Cliente"	4	5	6	7	8	9	Total general
"Cheque"	18	21	16	15	18	22	110
"Consignacion"	46	34	29	31	41	35	216
Total general	64	55	45	46	59	57	326

Contar de "Tipo Cliente"	Horas							
"Tipo Cliente"	4	5	6	7	8	9	10	Total general
"Cheque"	14	16	15	20	19	18		102
"Consignacion"	60	29	25	30	40	35	1	220
Total general	74	45	40	50	59	53	1	322







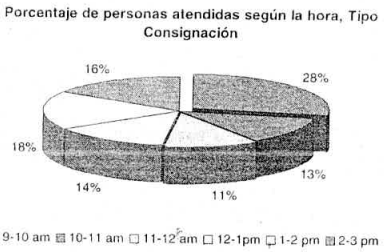


Figura 3.41 Número de clientes atendidos por los cajeros para los diferentes tipos de demanda y porcentajes de atención, para el caso discreto.

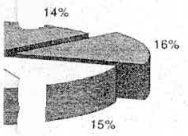
sto Cliente no Atendido	Costo Total
0.000	4.029.563
0.000	2.406.981
220.000	2.049.755
100.000	2.359.290

s /icio aleatorios.

)s rchivos *Excel*,
c de clientes, así
. Aquí se está ha-
l, ara facilitar el

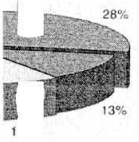
	6	7	8	9	10	Total general
	16	15	20	19	18	102
2	25	30	40	35	1	220
4	0	50	59	53	1	322

lidas según la hora, Tipo
ue



□12-1pm ▨1-2 pm □2-3 pm

lidas según la hora, Tipo
ación



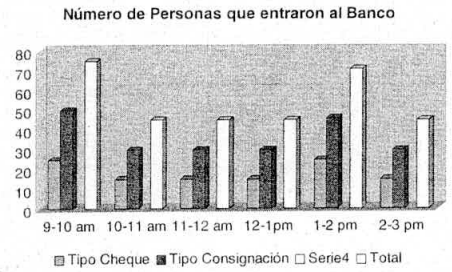
□12-1pm □1-2 pm ▨2-3 pm

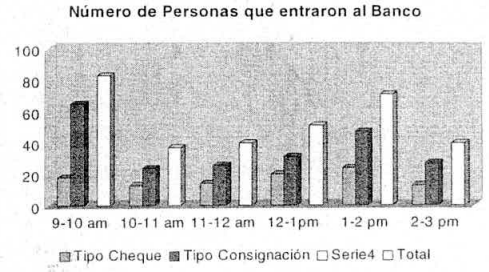
o. liferentes tipos
iscreto.

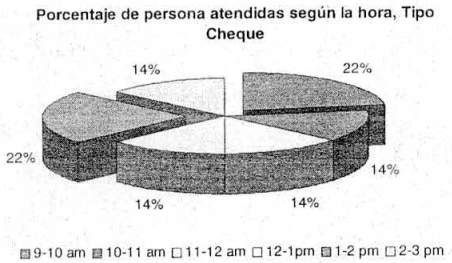
Gráficas con tiempos aleatorios

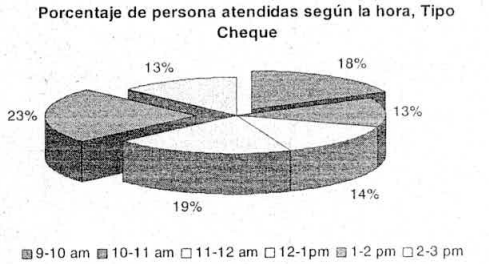
Contar de "Tipo Cliente"	Horas						
"Tipo Cliente"	4	5	6	7	8	9	Total general
"Cheque"	25	15	15	15	25	15	110
"Consignacion"	50	30	30	30	46	30	216
Total general	75	45	45	45	71	45	326

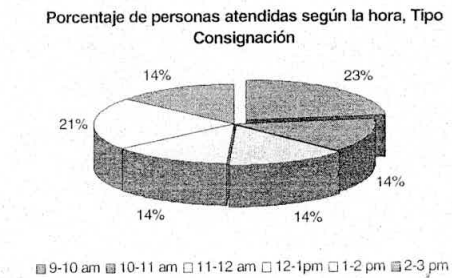
Contar de "Tipo Cliente"	Horas						
"Tipo Cliente"	4	5	6	7	8	9	Total general
"Cheque"	18	13	14	20	24	13	102
"Consignacion"	65	24	26	31	47	27	220
Total general	83	37	40	51	71	40	322











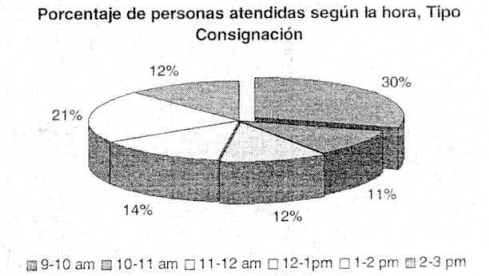


Figura 3.42 Número de clientes atendidos por los cajeros para los diferentes tipos de demanda y porcentajes de atención, para el caso aleatorio.

Gráficas de las llegadas de clientes y utilización de recursos con tiempos discretos

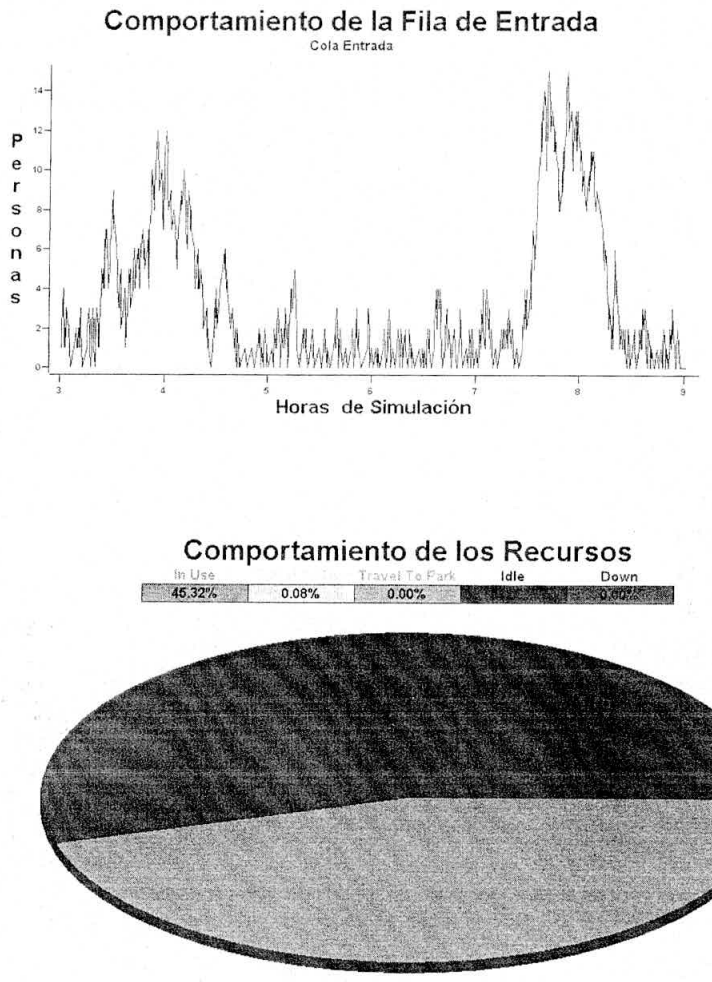


Figura 3.43 Gráficas *ProModel* de llegada de clientes y de utilización de los recursos para el caso discreto.

Gráficas de las llegadas de clientes y utilización de recursos con tiempos aleatorios

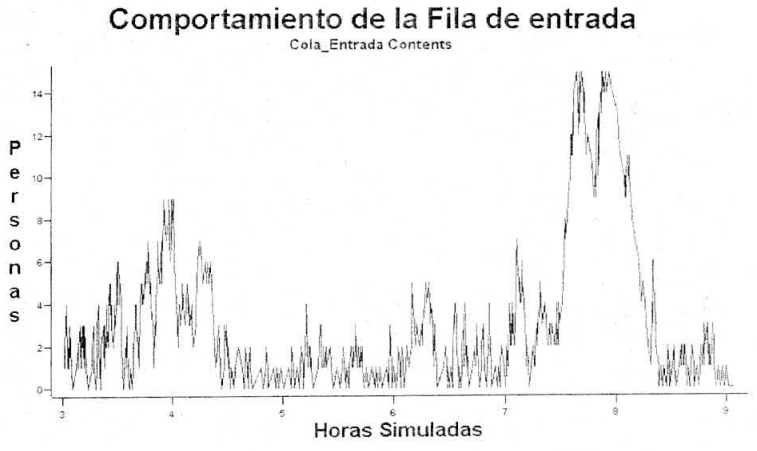


Figura 3.44 Gráficas *ProModel* de llegada de clientes y de utilización de los recursos para el caso aleatorio.

Caso 4. Sistemas *pull* versus sistemas *push*

El gerente de Industrias ECI, durante una reunión en el Club de Ejecutivos, escuchó que los sistemas de planeación y control de la producción *pull*, inventados por los japoneses, eran mejores que los sistemas *push,* creados por los estadounidenses, de manera que llamó a su ingeniero de producción, Juan Carlos Rodríguez, para que le diera un concepto sobre el tema.

Juan Carlos recordó inmediatamente que los inventarios se han usado, tradicionalmente, como colchones de ajuste contra la falta de materiales, cambios bruscos en la demanda, paradas de las máquinas y faltas de coordinación en la planeación de la producción; sin embargo, muchas veces el exceso de inventarios también puede causar mayores problemas que los originales de escasez, al congestionar el sistema y causar represamientos en el flujo de los materiales.

Más tarde, Juan Carlos revisó su biblioteca y encontró la historia del ingeniero japonés Taiichi Ohno, quien trabajó para Toyota coordinando la producción y la entrega de materiales y partes, con la producción y entrega de subensambles con los requerimientos de productos finales. El problema era complejo por la gran cantidad de componentes y por el elevado número de operaciones de un proceso de producción de automóviles. En un viaje a Estados Unidos, al observar el funcionamiento de los grandes supermercados, Ohno advirtió que el secreto de éstos estaba en conservar pequeñas cantidades de inventario (la de los exhibidores), de las que se surten los clientes, y pedir a los proveedores a medida que se agoten las existencias, de tal forma que el inventario siempre estaba a la vista y se encontraba limitado.

Ohno fue el primero en utilizar la palabra *pull* (halar) para indicar que es el cliente quien "hala" el sistema de inventarios. "Si el cliente no demanda, el inventario no aumenta". El sistema *pull* que desarrolló Ohno se basó en dos grandes ideas: el *Just In Time* (**JIT**) y la Automatización con Operarios Incluidos (*Autonomation*).

En su filosofía de producción *Just In Time* (Justo A Tiempo), Ohno imagina la producción como un flujo continuo de productos en proceso a través de operaciones, con inventarios en cada una de ellas, funcionando como "un supermercado de partes y subensambles", de manera que el producto que fluye va "tomando" las partes o subensambles necesarios en cada operación, y a medida que dichas partes se van consumiendo, se hacen pedidos a los proveedores, con el compromiso de que la reposición se haga en el instante preciso en que se necesita.

Además, se capacita a los operarios para manejar varias máquinas simultáneamente, se establecen equipos de trabajo y círculos de mejoramiento para estudiar y resolver los problemas relacionados con la calidad y se hace una inmensa inversión en tecnología: se diseñan sistemas de aprestamiento (*set up*) muy rápidos y a prueba de errores (*fool proof*) en las máquinas, se ubican las máquinas muy cerca unas de otras, formando una especie de U (celdas de manufactura), para minimizar el número de operarios y el número de desplazamientos de éstos, y se establecen sistemas de información visuales para parar la producción en caso de problemas, dentro de lo que Ohno llama *Autonomation*.

Revisando sus apuntes de clase, Juan Carlos encontró que el creador del sistema *Material Requeriments Planning* (**MRP**), prototipo de los sistemas *push* (empujar), fue Joseph Orlicky, quien como programador de **IBM** aprovechó las ventajas del manejo computarizado de bases de datos para diseñar una nueva forma de calcular los inventarios de productos con demanda dependiente, incluyendo además el tiempo como una nueva variable en el sistema.

El **MRP** es, entonces, un muy buen sistema de información porque calcula la lista de materiales necesarios para cada orden y el tiempo en el cual se debe terminar ésta. Posteriormente se desarrollaron sistemas **MRP** mucho más complejos, como el Mapics, el Vipics y otros conocidos. Estos son los sistemas **MRP II** (*Materials and Resources Planning*) que incluyen módulos para el cálculo de capacidades, costos y rutas de producción alternativas, hasta llegar a los más modernos que incluyen módulos de distribución y servicio al cliente, como el **DRP.** En esta época de globalización

halar) para indicar que
. "Si el cliente no de-
ue desarrolló Ohno se
a Automatización con

sto A Tiempo), Ohno
roductos en proceso a
de ellas, funcionando
de manera que el pro-
ambles necesarios en
consumiendo, se ha-
e que la reposición se

r varias máquinas si-
ulos de mejoramiento
con la calidad y se
sistemas de apresta-
ol proof) en las má-
otras, formando una
el número de opera-
ablecen sistemas de
de problemas, den-

ntró que el creador
prototipo de los sis-
no programador de
o de bases de datos
os de productos con
omo una nueva va-

nformación porque
n y el tiempo en el
ron sistemas **MRP**
s conocidos. Estos
anning) que inclu-
itas de producción
en módulos de dis-
a de globalización

y de empresas transnacionales globalizadas, totalmente intercomunicadas y coordinadas, los sistemas han evolucionado hacia los sistemas **ERP** (*Enterprise Resources Planning)* pero, internamente, continúan con la lógica del **MRP**.

Juan Carlos hojeó *La Carrera*[1], otro de los libros de Eliyahu Goldratt, denominado porque allí, le parecía recordar, se encontraba explicado otro de los sistemas *pull* denominado **DBR** (*Drum Buffer Rope*), que se puede traducir como tambor, inventario o colchón de seguridad, cuerda. En este prototipo de los sistemas logísticos de fabricación sincronizada, se trata de reducir la dispersión causada por las diferentes capacidades de las operaciones y por los niveles de inventario en un proceso de producción.

Las ideas centrales de este sistema consisten en buscar, primero que todo, las operaciones *cuellos de botella*, que son las más lentas y las "que marcan el paso" de la producción, para procurar que siempre trabajen al ritmo planeado, que semeja un tambor (*drum*), y asegurar que a éstas operaciones no les falten materias primas.

Esto se hace mediante un inventario de seguridad (*buffer*) que se mantiene siempre en las operaciones *cuellos de botella*. La cuerda (*rope*) que gráficamente une la pieza que se está fabricando en el *cuello de botella* con la última materia prima del inventario de seguridad no es otra cosa que un sistema para informar el instante en que debe entrar nueva materia prima al inventario, para que éste no disminuya.

La segunda idea importante es que el tambor (que marca el ritmo de producción) no debe "sonar" si no existe demanda para los productos. Sólo se produce cuando hay demanda. Así, como en el sistema **JIT**, el cliente "hala" la producción. Cada golpe de tambor equivale a una "autorización" para producir.

Juan Carlos imaginó que el "tamborilero" no podía ser otra persona que "el programador de la producción", quien debe organizar las órdenes de pedido de acuerdo con su fecha de entrega o en las secuencias previamente establecidas; preparar los materiales, organizar a los operarios, llenar los inventarios de seguridad y, finalmente, empezar a "tocar su tambor".

En su reciente lectura del libro *Factory Physics*[2], había visto otro sistema *pull* conocido bajo la denominación de sistema Conwip (*Constant Work*

1 Eliyahu Goldratt y Robert Fox, *La carrera*, NY, USA, North River Press, Inc.
2 Wallace Hopp y Mark Spearman, *Factory Physics, Foundations of Manufacturing Management*, Boston, Irwin-McGraw-Hill, 1996.

In Process), creado por Hopp y Spearman recogiendo lo más positivo de los Sistemas **JIT, DBR** y **MRP**.

La idea central de este sistema consiste en establecer un límite al inventario en proceso o **WIP** sin permitir entrada a ningún material o parte hasta que haya salido un producto terminado, esto es, se haya incrementado el *Throughput*. Lo primero que se debe hacer, en este caso, es definir "un nivel de inventario constante". Como se explicó en el Caso 1 de este capítulo, un valor sugerido será el nivel de inventario crítico, *Wo*.

Pero el problema de Juan Carlos para dar un concepto al gerente de Industrias ECI iba más allá de la simple definición de los dos sistemas. Se necesitaba la experimentación de los dos sistemas para compararlos, y la mejor forma de hacerlo era mediante simulación.

Parámetros que se deben considerar en la comparación

Juan Carlos nuevamente utilizó los parámetros de producción: *Throughput* (**TH**), o tasa a la cual la empresa produce partes vendibles, cuantificable en productos por unidad de tiempo; *Work In Process* (**WIP**) como cantidad de materiales y productos en proceso que se encuentran en el sistema, que se mide en unidades, y *Cycle Time* (**CT**) o tiempo que emplea un producto desde que ingresa al sistema hasta que sale, cuantificable en la unidad de tiempo con la que se esté trabajando, para la comparación entre los dos sistemas.

Modelando el sistema *MRP* (*push*)

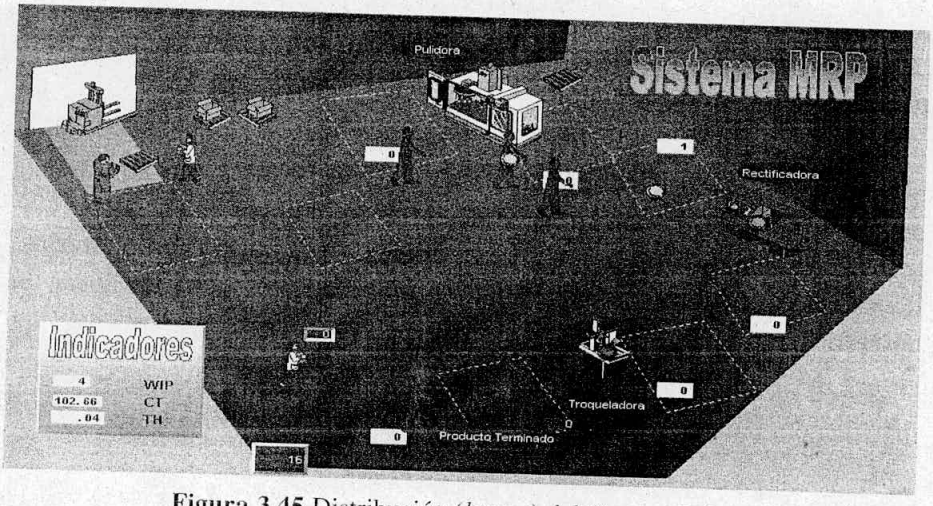


Figura 3.45 Distribución (*layout*) del Sistema MRP.

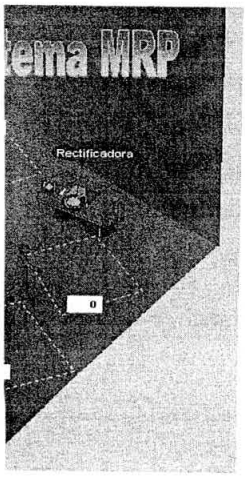
o lo más positivo de

ecer un límite al in-
gún material o parte
e haya incrementado
caso, es definir "un
Caso 1 de este capí-
co, *Wo*.

ıcepto al gerente de
los dos sistemas. Se
ra compararlos, y la

ıparación

ucción: *Throughput*
les, cuantificable en
P) como cantidad de
n el sistema, que se
ea un producto desde
nidad de tiempo con
os sistemas.



MRP.

1. *BUILD (Locations)*

Se consideraron 13 instalaciones: *recepción*, *pulidora*, *rectificadora*, *troqueladora*, *producto terminado*, *proveedor*, *consumidor*, además las tres máquinas tienen cada una *zona para materias primas* y *zona para producto semi-terminado*.

Los datos iniciales son:

Name	Capacity	Units	Down Times	Statistics	Rules
Pulidora	1	1	None	Time Series	Oldest
Rectificadora	1	1	None	Time Series	Oldest
Troqueladora	1	1	None	Time Series	Oldest
ZonaMat_dos	30	1	None	Time Series	Oldest
ZonaMat_tres	30	1	None	Time Series	Oldest
ZonaMat_cuatro	10	1	None	Time Series	Oldest
Producto Terminado	30	1	None	Time Series	Oldest
Proveedor	30	1	None	Time Series	Oldest
ZonaPT_dos	30	1	None	Time Series	Oldest
ZonaPT_tres	10	1	None	Time Series	Oldest
Consumidor	Infinite	1	None	Time Series	Oldest
ZonaEmbarque	Infinite	1	None	Time Series	Oldest
Recepción	Infinite	1	None	Time Series	Oldest

2. *BUILD (Entities)*

En el modelo se incluyeron siete entidades, cuatro de flujo de valor agregado: *pieza*, *pieza2*, *pieza3*, *pieza4* y tres de flujo de información: *orden*, *orden2*, y *orden_troqueladora*.

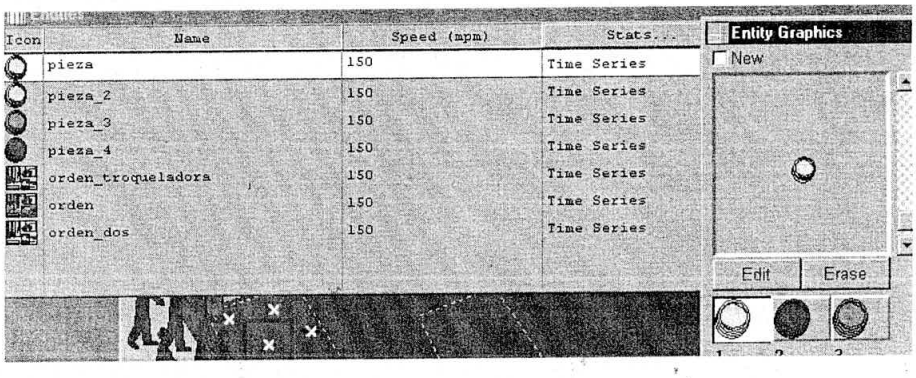


Figura 3.46 Entidades del modelo.

La entidad *pieza* llega a la primera instalación del proceso, que es *recepción*. Las demás entidades de flujo agregado (*pieza2*, *pieza3*, *pieza4*) son *piezas transformadas*.

3. *BUILD (Arrivals)*

Las llegadas se definieron así:

Entity	Location	Quantity	First Time	Occurrences	Frequency	Logic	Disable
Pieza	Recepción	1	0	Infinite	p(15)		no

4. *BUILD (Path network)*

Se definió una red de movimientos llamada *Desplazamientos*, con 53 trayectorias o *paths*, bidireccionales, que interconectan las instalaciones, y 13 *interfaces* que unen las instalaciones con los *nodos*. La información que se introdujo en el modelo es esta:

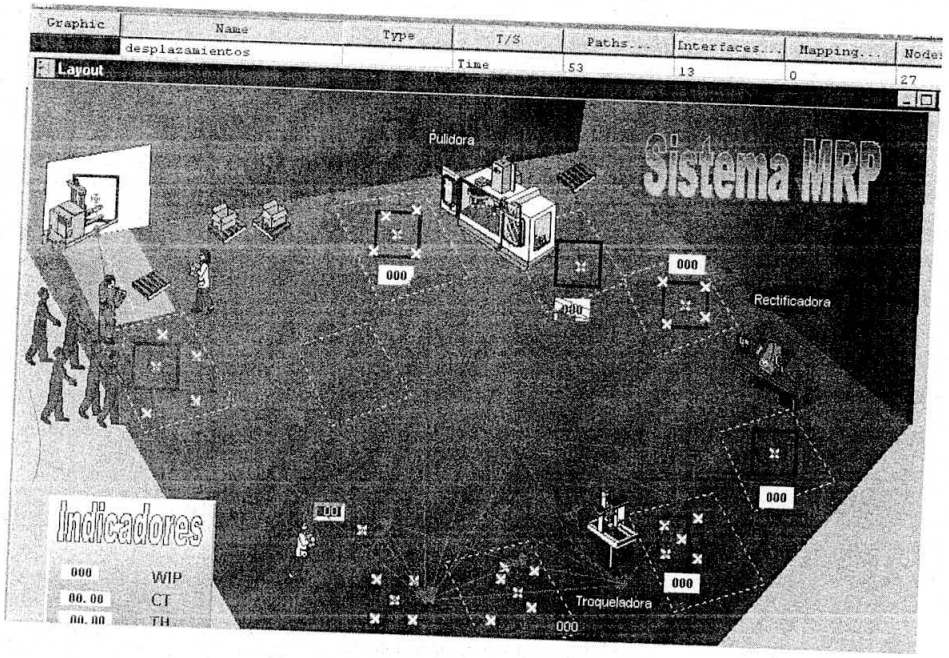


Figura 3.47 Red *desplazamientos*.

proceso, que es *re-*
a2, *pieza3*, *pieza4*)

quency	Logic	Disable
)		no

ientos, con 53 tra-
instalaciones, y 13
nformación que se

Interfaces	
Node	**Locations**
N3	ZonaMat_dos
N4	Pulidora
N5	ZonaMat_tres
N6	Rectificadora
N8	Troqueladora
N9	ProductoTerminado
N13	Proveedor
N15	ZonaPT_dos
N16	ZonaPT-tres
N19	ZonaEmbarque
N25	Consumidor
N27	Recepcion

5. *BUILD (Resources)*

En el modelo se utilizaron tres tipos de recursos: seis operarios, un montacargas y dos *operators*.

Name	Units	DTs	Statistics	Specifications	Home	Speed Empty	Speed Full	Closest Resource
Operario	6	None	By Unit	Desplazamientos	N1	N(100,20)	N(100,20)	Yes
Montacarga	1	None	By Unit	Desplazamientos	N13	150	150	Yes
Operator	2	None	By Unit	Desplazamientos	N1	E(100)	N(50,20)	Yes

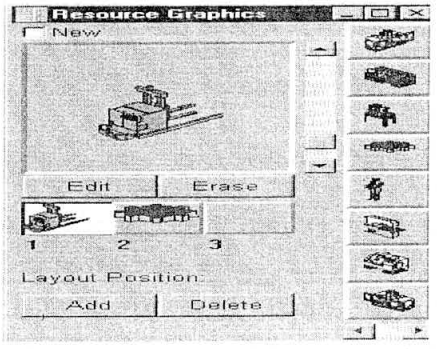


Figura 3.48 El recurso *Montacargas* tiene dos gráficos.

Al hacer clic en el campo *Logic*, se colocó la siguiente información:

Resource	Node	Entry Logic	Exit Logic
Montacarga	N13	GRAPHIC 2	GRAPHIC 1

Esto hace que el recurso *Montacarga* cuando está en el *Nodo 13* (*Proveedor*). En el momento de llegar, utilice la *Gráfica 1*; cuando salga, use la *Gráfica 2*, que se definen quitando el chulo de *New* en la ventana *Resources Graphics*.

6. *BUILD (Attributes)*

Se utilizaron dos atributos, con la siguiente información:

Identification	Type	Classification
Att	Real	Entity
Att_orden	Real	Entity

7. *BUILD (Variables)*

Además, se incluyeron catorce variables, con la siguiente información:

Variables (global)

Icon	ID	Type...	Initial value
Yes	ct	Real	0
Yes	th	Real	0
No	qzona_cuatro	Integer	0
No	ordenes	Integer	0
No	ordenes_troqueladora	Integer	0
No	qtroqueladora	Integer	0
No	ordenes_rectificadora	Integer	0
No	qrectificadora	Integer	5
No	ordenes_pulidora	Integer	0
No	wip2	Integer	0
Yes	wip	Integer	0
No	wip3	Integer	0
No	wip4	Integer	0
No	CT_orden	Real	0

Figura 3.49 Variables del modelo.

8. *BUILD (Subroutine)*

Se utilizó la subrutina *cal_th* para calcular el *Throughput*, definida de la siguiente manera:

Identification	Type	Parameter	Logic
Cal_th	None		If (ct<>0) then th=wip/ct

9. *BUILD (More elements-External files)*

También se trabajó con un archivo externo llamado *archivo* para escribir en *resultados.txt*, definido así:

Identification	Type	File Name	Prompt
archivo	General Write	C:\Temp \MRP\resultados.txt	

10. *BUILD (Processing)*

Como el proceso es extenso, se comentarán las instrucciones en grupos de cuatro:

Entity	Location	Operation	Output	Destination	Rule	Move Logic
All	Recepción	**INC** wip Att=**CLOCK()** Cal_th()	All	ZonaMat_dos	First1	**MOVE WITH** Operario **OR** Operator **THEN** **FREE**
All	ZonaMat_dos	Cal_th()	Pieza2	Pulidora	First1	**MOVE WITH** Operario **OR** Operator **THEN** **FREE**
All	Pulidora	Cal_th() **WAIT** e(10) **INC** wip3	Pieza2	ZonaPT_dos	First1	**MOVE WITH** Operario **OR** Operator **THEN** **FREE**
All	ZonaPT_dos	Cal_th()	Pieza2	ZonaMat_tres	First1	**MOVE WITH** Operario **OR** Operator **THEN** **FREE**

En la primera instrucción, cualquier entidad que llegue al sistema hace incrementar la variable *Wip* o sea el inventario en proceso; se guarda el tiempo de entrada de la entidad en el atributo *Att* y se calcula el *Throughput*, mediante la subrutina *Cal_th()*. La entidad que entra sale con destino a la instalación *ZonaMat_dos* en el mismo orden de llegada, es transportada por el recurso *Operario* o por el recurso *Operator* y, tan pronto llegan a la instalación *ZonaMat_dos*, el recurso regresa a su base.

En la segunda instrucción, la entidad entra en la instalación *ZonaMat_dos*. Se calcula el *Throughput* mediante la subrutina *Cal_th()*; la

entidad que sale es *pieza2* con destino a la instalación *Pulidora*, transportada por el recurso que esté disponible entre *Operador* u *Operator*.

En la tercera instrucción la entidad *Pieza2* entra en la instalación *Pulidora*, se calcula el *Throughput* mediante la subrutina *Cal_th()*; espera un tiempo con distribución exponencial con media 10, incrementa el inventario en proceso de la instalación 3, que se guarda en la variable *Wip 3* y sale con destino a la instalación *ZonaPT_dos* llevada por el recurso disponible.

En la cuarta instrucción *pieza 2* entra a la instalación *ZonaPT_dos;* se calcula el *Throughput* mediante la subrutina *Cal_th()* y sale con destino a la instalación *ZonaMat_tres*, llevada por el recurso disponible.

Entity	Location	Operation	Output	Destination	Rule	Move Logic
All	ZonaMat_tres	Cal_th()	Pieza3	Rectificadora	First1	**MOVE WITH** Operario **OR** Operator **THEN FREE**
All	Rectificadora	**DEC** wip3 **WAIT** n(20,10) **INC** wip2 Cal_th()	Pieza3	ZonaPT_tres	First1	**MOVE WITH** Operario **OR** Operator **THEN FREE**
All	ZonaPT_tres	Cal_th()	Pieza3	ProductoTermina do	First1	**MOVE WITH** Operario **OR** Operator **THEN FREE**
All	ProductoTermina do	Cal_th()	Pieza4	Troqueladora	First1	**MOVE WITH** Operario **OR** Operator **THEN FREE**

En estas cuatro instrucciones no hay nada nuevo, distinto de observar que, al pasar una entidad de una operación a otra, se decrementa el *WIP* de la anterior y se incrementa el *WIP* de la siguiente, y que la entidad se va transformando: *pieza2* en *pieza3*, etcétera.

Entity	Location	Operation	Output	Destination	Rule	Move Logic
All	Troqueladora	**DEC** wip2 **WAIT** e(15) **INC** wip4 Cal_th() Ct=**CLOCK**()-Att	Pieza4	ZonaEmbarque	First1	**MOVE WITH** Operario **OR** Operator **THEN FREE**
All	ZonaEmbarque	Cal_th()	Pieza4	Producto_terminado	First1	**MOVE WITH** Operario **OR** Operator **THEN FREE**
All	Producto_terminado	**DEC** wip4 Cal_th() CT_orden=**CLOCK**() -Att_orden	Orden Troqueladora	Consumidor	First1	**MOVE WITH** Operario **OR** Operator **THEN FREE** **DEC** wip
All	Consumidor	**WAIT** 300hr	All	EXIT	First1	**MOVE WITH** Operario **OR** Operator **THEN FREE**

En este grupo vale la pena comentar que, en la tercera instrucción, se calcula el *tiempo de ciclo de la orden* tomando la hora del reloj y restándole el contenido del *Att_orden*.

11. *SIMULATION (Options)*

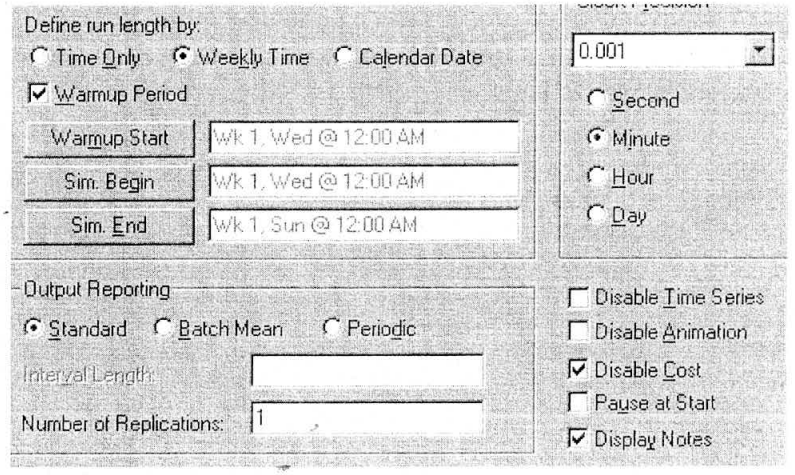


Figura 3.50 Ventana de opciones para la simulación.

Utilizando el *Browse* se guarda en *C:\Temp*, se marca *Weekly Time*, se define la hora de iniciación, 12:00 AM del miércoles de la semana 1, y la de terminación, 12:00 AM del viernes de la semana 5. Luego se da *Run* para iniciar la simulación.

VARIABLES

Variable Name	Total Changes	Average Minutes Per Change	Minimum Value	Maximum Value	Current Value	Average Value
ct	281	20.49	0	1418.31	1387.92	645.35
th	3406	1.69	0	0.20	0.06	0.06
qzona cuatro	0	0.00	0	0	0	0
ordenes	0	0.00	0	0	0	0
ordenes troqueladora	0	0.00	0	0	0	0
qtroqueladora	0	0.00	0	0	0	0
ordenes rectificadora	0	0.00	0	0	0	0
qrectificadora	0	0.00	5	5	5	5
ordenes pulidora	0	0.00	0	0	0	0
wip2	564	10.21	0	4	0	0.42
wip	653	8.81	0	96	91	45.06
wip3	627	9.18	0	61	61	35.20
wip4	562	10.24	0	2	0	0.03
CT orden	281	20.49	0	5759.04	5759.04	2863.82

Figura 3.51 Resultados de la simulación con MRP.

Fácilmente se observa que, en este sistema **MRP** prototipo de los modelos *push*, el inventario en proceso crece bastante y que el tiempo de ciclo también aumenta, mientras que el *throughput* permanece constante, con una ligera tendencia a la baja.

ırca *Weekly Time*, se
de la semana 1, y la
5. Luego se da *Run*

imum alue	Current Value	Average Value
3.31	1387.92	645.35
0.20	0.06	0.06
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
5	5	5
0	0	0
4	0	0.42
96	91	45.06
61	61	35.20
2	0	0.03
0.04	5759.04	2863.82

MRP.

rototipo de los mo-
e el tiempo de ciclo
ece constante, con

Modelando el sistema *JIT o Kanban (pull)*

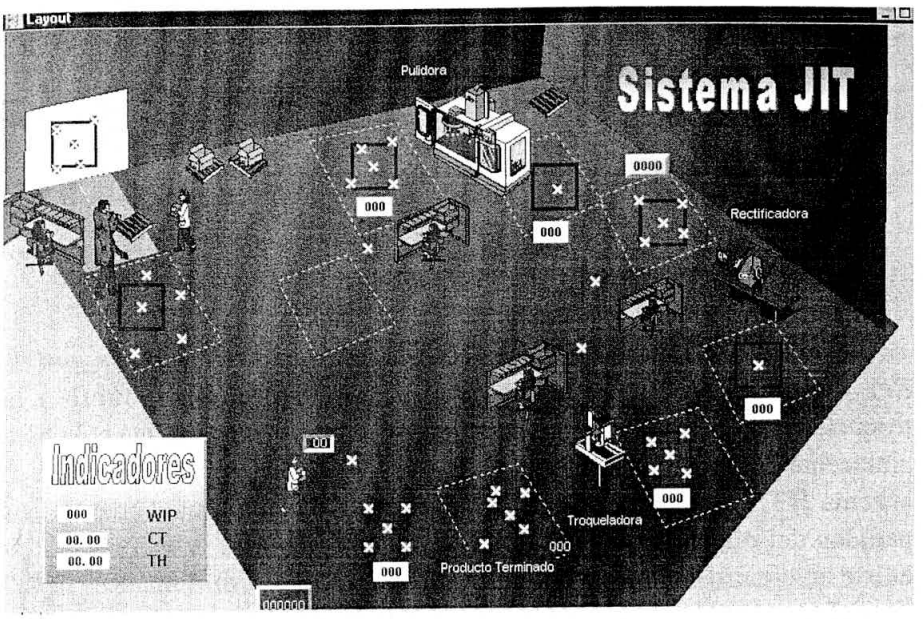
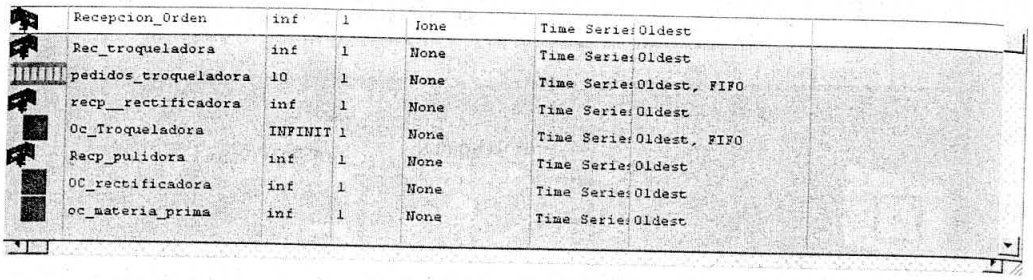


Figura 3.52 Distribución (*layout*) del Sistema JIT o KANBAN.

Este es básicamente el mismo modelo anterior, con algunas adiciones.

1. *BUILD (Locations)*

Name	Capacity	Units	Down Times	Statistics	Rules
Pulidora	1	1	None	Time Series	Oldest
Rectificadora	1	1	None	Time Series	Oldest
Troqueladora	1	1	None	Time Series	Oldest
ZonaMat_dos	30	1	None	Time Series	Oldest
ZonaMat_tres	30	1	None	Time Series	Oldest
ZonaMat_cuatro	10	1	None	Time Series	Oldest
Producto_Terminado	30	1	None	Time Series	Oldest
Proveedor	30	1	None	Time Series	Oldest
ZonaPT_dos	30	1	None	Time Series	Oldest
ZonaPT_tres	10	1	None	Time Series	Oldest
Consumidor	Infinite	1	None	Time Series	Oldest
ZonaEmbarque	Infinite	1	None	Time Series	Oldest
Recepción	Infinite	1	None	Time Series	Oldest



	Recepcion_Orden	inf	1	None	Time Series:Oldest
	Rec_troqueladora	inf	1	None	Time Series:Oldest
	pedidos_troqueladora	10	1	None	Time Series:Oldest, FIFO
	recp__rectificadora	inf	1	None	Time Series:Oldest
	Oc_Troqueladora	INFINIT	1	None	Time Series:Oldest, FIFO
	Recp_pulidora	inf	1	None	Time Series:Oldest
	OC_rectificadora	inf	1	None	Time Series:Oldest
	oc_materia_prima	inf	1	None	Time Series:Oldest

Figura 3.53 Instalaciones adicionales del modelo.

Se incluyeron cuatro sitios para colocar las órdenes: *Recepción_orden*, *Rec_troqueladora*, *Recp_rectificadora* y *Recp_pulidora*. Junto a la troqueladora, que es la última operación del proceso, se puso una cola para almacenar los pedidos u órdenes, llamada *Pedidos_troqueladora*. En el sistema **JIT**, como se mencionó, la demanda es "halada" por el cliente. Los pedidos del cliente se ubican en la instalación *Pedidos_troqueladora*; si existe inventario en la instalación *Producto terminado*, el cliente se "lleva" su pedido y se incrementa el *Throughput*. Las instalaciones *Oc_troqueladora*, *Oc_rectificadora* y *Oc_materia prima* son sólo marcas (cuadrado rojo con X blanca) de los sitios donde se deben ubicar los materiales.

Si no existe inventario, se coloca una "orden" en el sitio *Rec_troqueladora*, que corresponden a la *Troqueladora*, autorizando a esta operación para que haga un producto. Si en la *ZonaPT_cuatro*, sitio donde se coloca el producto terminado de la *Troqueladora*, existe inventario, éste pasa a *Producto terminado*; si no, se genera una *Orden* que se coloca en *Recp_rectificadora*.

De la misma forma, si en la *ZonaPT_tres,* que corresponde a la *Rectificadora*, existe inventario, éste pasa a la *Troqueladora*, si no, se genera una "orden" que se coloca en el sitio *Recp_pulidora*. Nuevamente, si en la *ZonaPT_dos*, que es la de la "pulidora", no existe inventario, se genera una *Orden* en *Recepción_orden* para que envíe materias primas.

Es fácil entender que si se dejan vacíos los almacenamientos de las diferentes operaciones, la atención a una orden del cliente se va a demorar todo el tiempo de ciclo, o sea el tiempo necesario en pedir los materiales, pulir, rectificar, troquelar y enviar a producto terminado. En la analogía con el supermercado, es como si éste se encontrase vacío en el momento de la llegada de los clientes. De manera que, en las zonas de producto terminado, "deberá existir un inventario de seguridad", que es el WIP.

odelo.

: *Recepción_orden*,
lidora. Junto a la
puso una cola para
roqueladora. En el
" por el cliente. Los
os_troqueladora; si
el cliente se "lleva"
iones *Oc_troquela-*
marcas (cuadrado
los materiales.

n el sitio *Rec_tro-*
zando a esta opera-
tro, sitio donde se
ste inventario, éste
n que se coloca en

corresponde a la
dora, si no, se ge-
ora. Nuevamente,
iste inventario, se
materias primas.

enamientos de las
te se va a demorar
dir los materiales,
lo. En la analogía
en el momento de
le producto termi-
s el WIP.

2. *BUILD (Entities)*

Icon	Name	Speed (mpm)	Stats...
	pieza	150	Time Series
	pieza_2	150	Time Series
	pieza_3	150	Time Series
	pieza_4	150	Time Series
	orden_pulidora	150	Time Series
	orden_rectificadora	150	Time Series
	orden_troqueladora	150	Time Series
	pedido	50	Time Series

Figura 3.54 Entidades del modelo.

3. *BUILD (Path Networks)*

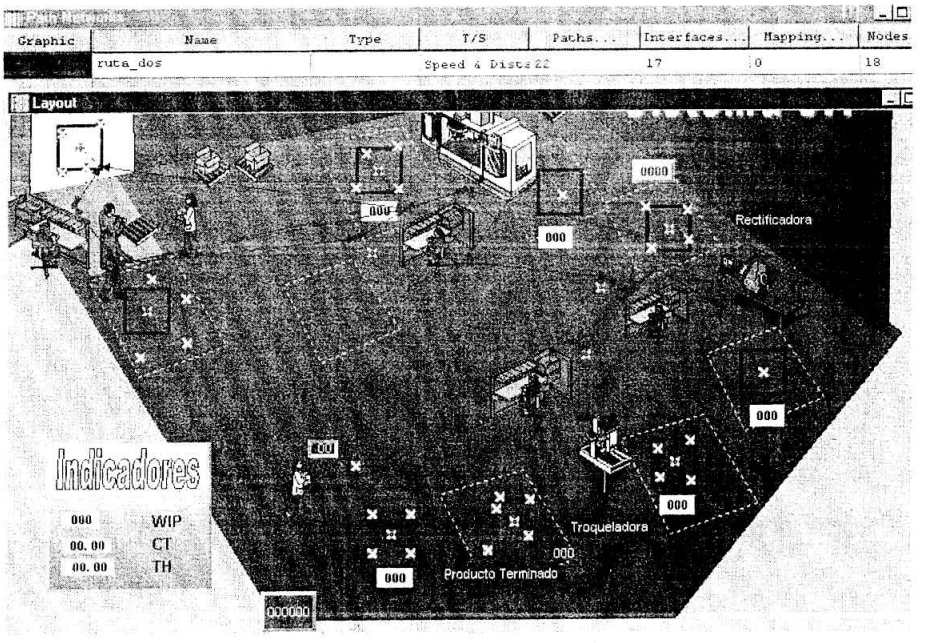


Figura 3.55 Red *Ruta_dos* del modelo.

Interfaces	
Node	**Locations**
N1	Recepción_orden
N2	Rec_troqueladora
N3	Oc_troqueladora
N4	Recp_rectificadora
N5	Oc_rectificadora
N6	Recp_pulidora
N7	ZonaMat_cuatro
N9	Rectificadora
N10	ZonaMat_tres
N11	Pulidora
N12	ZonaMat_dos
N13	Recepción
N14	Proveedor
N15	Oc_materia prima
N16	ProductoTerminado
N17	ZonaEmbarque
N18	Troqueladora

Variables (global) [18]

Icon	ID	Type...	Initial value	Stats...	Notes...
Yes	ct	Real	1	Time Series	
Yes	th	Real	0	Time Series	
No	qzona_cuatro	Integer	0	Time Series	suponiendo que t
No	ordenes	Integer	0	Time Series	
No	ordenes_troqueladora	Integer	0	Time Series	
No	qtroqueladora	Integer	0	Time Series	
No	ordenes_rectificadora	Integer	0	Time Series	
No	qrectificadora	Integer	5	Time Series	
No	ordenes_pulidora	Integer	0	Time Series	
No	wip2	Integer	0	Time Series	
Yes	wip	Integer	0	Time Series	
No	wip3	Integer	0	Time Series	
No	wip4	Integer	0	Time Series	
No	CT_orden	Real	0	Time Series	
No	q_conwip	Integer	50	Time Series	
No	th_conwip	Integer	0	Time Series	
No	hora	Integer	0	Time Series	

Attributes [3]

ID	Type...	Classification...	Notes...
att	Real	Ent	
att_orden	Real	Ent	
temp		Ent	

Resources [3]

Icon	Name	Units	DTs...	Stats...	Specs...	Search...	Logic...	Pts/...	Notes...
	Operario	1	None	By Unit	ruta_dos, I	None	0	1	
	Montacarga	1	None	By Unit	No Network	None	0	0	
	Operator	1		Unit	ruta_dos, I	None	0	1	

Arrivals [1]

Entity...	Location...	Qty each...	First Time	Occurrences	Frequency	Logic	Disable
pedido	Rec_troqueladora	10	0	1	1		No
pieza	zona_cuatro	1	0	2	1		Yes
pieza_2	zona_tres	1	0	1	1		Yes
pieza_3	Zona_dos	1	0	1	1		Yes
pieza_4	Proveedor	50	Wk 1, Wed @ 12:	1	1		No

Figura 3.56 Variables, atributos, recursos y llegadas del modelo.

Nota. En *las locations Zona_cuatro, zona_tres, zona_dos*, escribir *ZonaMat_cuatro*, etc.

Subroutines

ID	Type...	Parameters...
cal_th		None

Logic

```
if (ct<>0) then
th=wip/ct
```

sub_wip		None

Logic

```
wip =contents(zona_cuatro)+contents(zona_tres)+contents(zona_dos)
+contents(rectificadora)+contents(troqueladora)+contents(pulidora)
th=wip/ct
```

Figura 3.57 Subrutinas del modelo.

4. *BUILD (Processing)*

Entity	Location	Operation	Output	Destination	Rule	Move Logic
pedido	Rec_troqueladora	Sub_wip()	pedido	Pedidos_troqueladora	First 1	
pedido	Pedidos_troqueladora	IF (CONTENTS(ZonaMat_cuatro) <=2) THEN { ORDER 1 pedido TO Oc_Troqueladora } sub_wip()	pedido	ZonaMat_cuatro	JOIN 1	
pieza	ZonaMat_cuatro	sub_wip() JOIN 1 pedido	pieza_2	Troqueladora	First 1	MOVE WITH Operario THEN FREE
pieza_2	Troqueladora	sub_wip() WAIT 5 min	pieza_2	ZonaEmbarque	First 1	ct = CLOCK(MIN) - temp MOVE WITH Operario THEN FREE
pieza_2	ZonaEmbarque	sub_wip()	pieza_2	Producto_Terminado	First 1	MOVE WITH Operario THEN FREE
pieza_2	Producto_Terminado	sub_wip() WAIT 5hr	pieza_2	EXIT	First 1	

En este grupo de instrucciones se observa que la entidad *pedido*, que es una orden de material, se coloca en la recepción de órdenes de la troqueladora *Rec_troqueladora*. La subrutina *sub_wip()* calcula el WIP del sistema. El *pedido* se coloca en la cola *Pedidos_troqueladora*. En este modelo se ha supuesto que se deben mantener por lo menos dos unidades de productos en proceso en las zonas de almacenamiento de materiales. Si el contenido de la *ZonaMat_cuatro* es menor que o igual a dos, se coloca un "pedido" en la recepción de órdenes de la operación anterior, o sea en "Oc_troqueladora". El *pedido* sigue su ruta hacia *ZonaMat_ cuatro* y allí se junta con una *pieza* mediante la instrucción JOIN 1. Con este comando, el *pedido* únicamente "regresa" si encuentra material disponible; de lo contrario, se queda esperando; de manera que se pueden "almacenar" varios *pedidos*. Al salir se tiene una nueva entidad: *pieza_2*. La nueva entidad sigue hacia la *Troqueladora*, donde se demora cinco minutos en procesarse y sigue hacia *ZonaEmbarque*. En el espacio donde se define la regla de movimiento, se calcula el tiempo de ciclo. Después la pieza va a *Producto_Terminado*. En todo momento se verifica la cantidad de inventario en proceso, con el fin de que éste no se agote.

Entity	Location	Operation	Output	Destination	Rule	Move Logic
pedido	Oc_troqueladora	sub_wip()	pedido	Recp_rectificadora	First 1	MOVE WITH Operator THEN FREE

pedido	Recp_rectificadora	sub_wip()
		IF (CONTENTS(ZonaMat_tres) <=2) THEN
		{
		ORDER 1 pedido TO Oc_rectificadora
		}
			pedido	ZonaMat_tres	JOIN 1	

pedido	ZonaPT_tres	sub_wip() WAIT 5hr	pedido	EXIT	First 1	
pieza_2	ZonaMat_tres	sub_wip() JOIN 1 pedido	pieza	Rectificadora	First 1	MOVE WITH Operator THEN FREE
pieza	Rectificadora	sub_wip() wait 10 min	pieza	ZonaMat_cuatro	First 1	MOVE WITH Operator THEN FREE
pedido	Oc-rectificadora	sub_wip()	pedido	Recp_pulidora	First 1	MOVE WITH Operator THEN FREE
pedido	Pulidora	sub_wip()	pedido	EXIT	First 1	

Este grupo de instrucciones es muy similar al anterior.

entidad *pedido*, que es
lenes de la troqueladora
el WIP del sistema. El
En este modelo se ha
unidades de productos
eriales. Si el contenido
coloca un "pedido" en
en "Oc_troqueladora".
se junta con una *pieza*
el *pedido* únicamente
ıtrario, se queda espe-
s *pedidos*. Al salir se
tidad sigue hacia la
ocesarse y sigue hacia
gla de movimiento, se
ducto_Terminado. En
en proceso, con el fin

	Rule	Move Logic
id	First 1	MOVE WITH Operator THEN FREE
	JOIN 1	
	First 1	
	First 1	MOVE WITH Operator THEN FREE
	First 1	MOVE WITH Operator THEN FREE
	First 1	MOVE WITH Operator THEN FREE
	First 1	

ior.

Entity	Location	Operation	Output	Destination	Rule	Move Logic
pedido	Recp_pulidora	IF(CONTENTS(ZonaMat_dos) <=4) THEN { ORDER 1 pedido TO Oc_materia_prim a }	pedido	ZonaMat_dos	JOIN 1	
pieza_3	ZonaMat_dos	JOIN 1 pedido	pieza_2	Pulidora	First 1	MOVE WITH Operario THEN FREE
pieza_2	Pulidora	sub_wip() WAIT 3min	pieza_2	ZonaMat_tres	First 1	MOVE WITH Operario THEN FREE
pedido	Oc_materia_prima	sub_wip()	pedido	Recepción	First 1	MOVE WITH Operario THEN FREE
pedido	Recepción	sub_wip()	pedido	Proveedor	JOIN 1	
pieza_4	Proveedor	JOIN 1 pedido temp=CLOCK(MIN)	pieza_3	ZonaMat_dos	First 1	MOVE WITH Operario THEN FREE

En este grupo las instrucciones también son similares.

5. *SIMULATION (Options)*

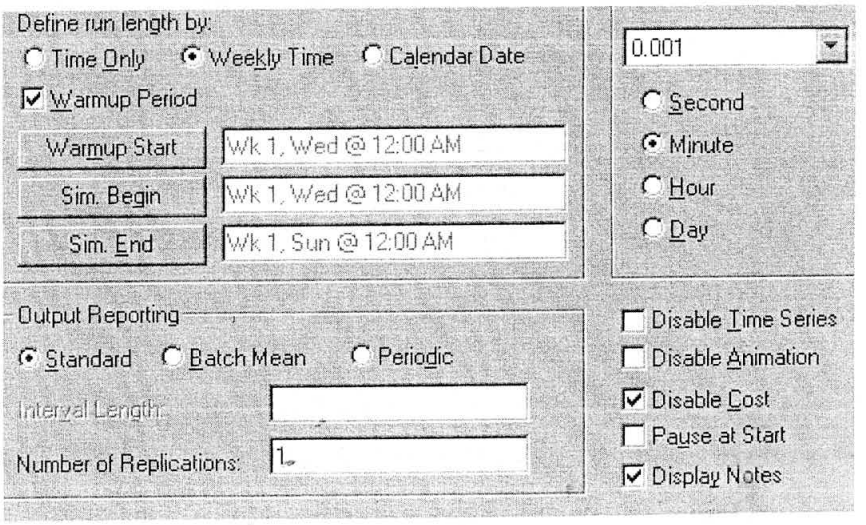


Figura 3.58 Tiempo de simulación.

General Report

Name	Changes	Per Change	Value	Value	Value	Value
ct	10	15.72	1	125.33	125.33	123.45
th	140	1.12	0	7	0	0.02
qzona cuatro	0	0.00	0	0	0	0
ordenes	0	0.00	0	0	0	0
ordenes troqueladora	0	0.00	0	0	0	0
qtroqueladora	0	0.00	0	0	0	0
ordenes rectificadora	0	0.00	0	0	0	0
qrectificadora	0	0.00	5	5	5	5
ordenes pulidora	0	0.00	0	0	0	0
wip2	0	0.00	0	0	0	0
wip	140	1.12	0	10	0	0.13
wip3	0	0.00	0	0	0	0
wip4	0	0.00	0	0	0	0

Figura 3.59 Resultados de la simulación JIT.

Si se observan y se comparan los resultados del **JIT** con los del **MRP**, hay una notable reducción de los inventarios en proceso y del tiempo de ciclo, con un ligero decremento en el *throughput*.

Modelando el sistema *CONWIP (pull)*

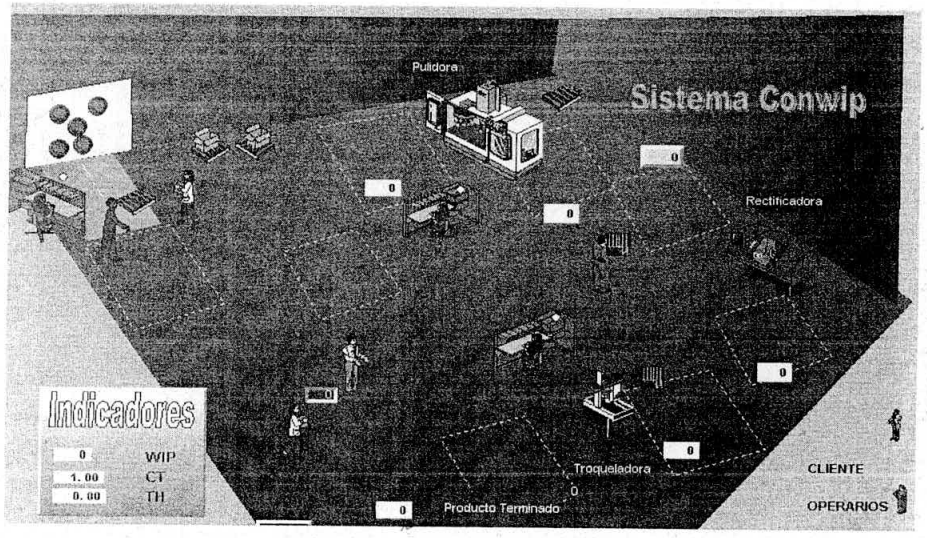


Figura 3.60 Distribución (*layout*) del sistema CONWIP.

lue	Value	Value
.33	125.33	123.45
7	0	0.02
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
5	5	5
0	0	0
0	0	0
10	0	0.13
0	0	0
0	0	0

JIT.

T con los del **MRP**, eso y del tiempo de

ONWIP.

Este modelo es básicamente el mismo del **JIT.** Las *locations*, las *entities*, la *path network* son las mismas. Se hicieron algunos cambios:

a) BUILD (Resources)

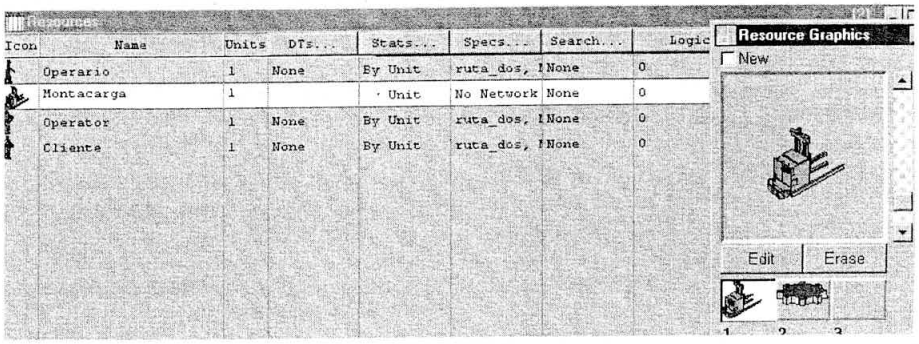


Figura 3.61 Recursos del modelo.

b) BUILD (Arrivals)

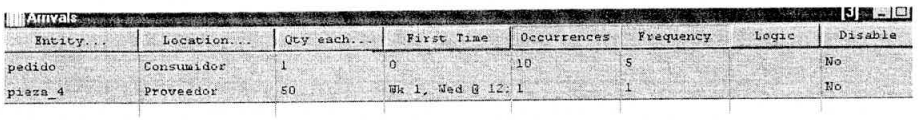


Figura 3.62 Llegadas al modelo.

c) *BUILD (Processing)*

Entity	Location	Operation	Output	Destination	Rule	Move Logic
pedido	Rec_troqueladora	Sub_wip()	pedido	Pedidos_troqueladora	First 1	
pedido	Pedidos_troqueladora	IF (CONTENTS(ZonaMat_cuatro) <=2) THEN { ORDER 1 pedido TO Oc_Troqueladora } sub_wip()	pedido	ZonaMat_cuatro	JOIN 1	
pieza	ZonaMat_cuatro	sub_wip() JOIN 1 pedido	pieza_2	Troqueladora	First 1	MOVE WITH Operario THEN FREE
pieza_2	Troqueladora	sub_wip() WAIT 10 min	pieza_2	ZonaEmbarque	First 1	ct = CLOCK(MIN) - temp MOVE WITH Operario THEN FREE
pieza_2	ZonaEmbarque	sub_wip()	pieza_2	Producto_Terminado	First 1	MOVE WITH Operario THEN FREE
pieza_2	Producto_Terminado	sub_wip() WAIT 5hr	pieza_2	EXIT	First 1	
pedido	Oc_troqueladora	sub_wip()	pedido	Recp_rectificadora	First 1	MOVE WITH Operator THEN FREE
pedido	ZonaPT_tres	sub_wip() WAIT 5hr	pedido	EXIT	First 1	
pieza_2	ZonaMat_tres	sub_wip()	pieza	Rectificadora	First 1	MOVE WITH Operator THEN FREE
pieza	Rectificadora	sub_wip()	pieza	ZonaMat_cuatro	First 1	MOVE WITH Operator THEN FREE
pedido	Pulidora	sub_wip()	pedido	EXIT	First 1	

ıtion	Rule	Move Logic
_tro ra	First 1	
ıt_cu	JOIN 1	
ıdor	First 1	MOVE WITH Operario THEN FREE
bar	First 1	ct = CLOCK(MIN) - temp MOVE WITH Operario THEN FREE
_Te	First 1	MOVE WITH Operario THEN FREE
	First 1	
tifi	First 1	MOVE WITH Operator THEN FREE
	First 1	
or	First 1	MOVE WITH Operator THEN FREE
cu	First 1	MOVE WITH Operator THEN FREE
	First 1	

Este grupo de instrucciones es similar a las del **JIT.**

Entity	Location	Operation	Output	Destination	Rule	Move Logic
pedido	Recp_pulidora	IF (CONTENTS(ZonaMat_dos) <=4) THEN { ORDER 1 pedido TO Oc_materia_prima }	pedido	ZonaMat_dos	JOIN 1	
pieza_3	ZonaMat-dos	JOIN 1 pedido	pieza_2	Pulidora	First 1	MOVE WITH Operator THEN FREE
pieza_2	Pulidora	sub_wip()	pieza_2	ZonaMat_tres	First 1	MOVE WITH Operator THEN FREE
pedido	Oc_materia_prima	sub_wip()	pedido	Recepción	First 1	MOVE WITH Operator THEN FREE
pedido	Recepción	sub_wip()	pedido	Proveedor	JOIN 1	
pieza_4	Proveedor	JOIN 1 pedido temp=CLOCK(MIN)	pieza_3	ZonaMat-dos	First 1	MOVE WITH Operator THEN FREE
pedido	Consumidor		pedido	Rec_troqueladora	First 1	MOVE WITH Operator THEN FREE

Se asume que el *Wip* del sistema no supera cuatro; cada vez que se baje de dicho nivel, se ordena entre una *pieza* a *recepción.*

VARIABLES

Variable Name	Total Changes	Average Minutes Per Change	Minimum Value	Maximum Value	Current Value	Average Value
ct	10	13.64	1	87.72	87.72	86.65
th	120	1.14	0	3	0	0.00
qzona cuatro	0	0.00	0	0	0	0
ordenes	0	0.00	0	0	0	0
ordenes troqueladora	0	0.00	0	0	0	0
qtroqueladora	0	0.00	0	0	0	0
ordenes rectificadora	0	0.00	0	0	0	0
qrectificadora	0	0.00	5	5	5	5
ordenes pulidora	0	0.00	0	0	0	0
wip2	0	0.00	0	0	0	0
wip	120	1.14	0	8	0	0.09
wip3	0	0.00	0	0	0	0
wip4	0	0.00	0	0	0	0
CT orden	0	0.00	0	0	0	0
q conwip	0	0.00	50	50	50	50
th conwip	0	0.00	0	0	0	0
hora	0	0.00	0	0	0	0

Figura 3.63 Resultados de la simulación con CONWIP.

El tiempo de ciclo promedio y los inventarios en proceso son menores que los de **JIT**, mientras que el *throughput* es el mismo.

Modelando el Sistema *DBR (pull)*

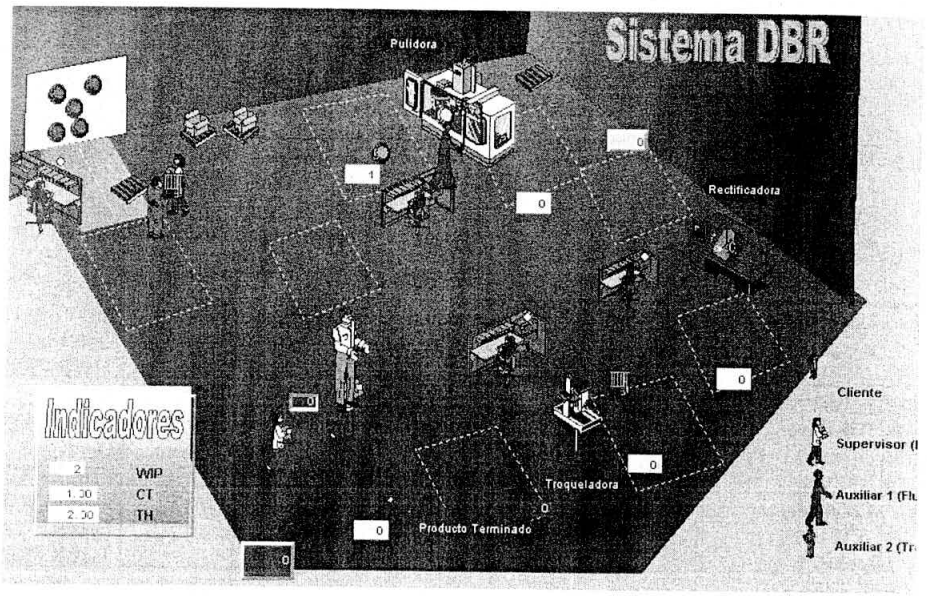


Figura 3.64 Distribución (*layout*) del sistema DRB.

Este modelo es básicamente el modelo **JIT.** Las *locations, entities, path network, variables, attributes, subroutines* son iguales. Los cambios son los siguientes:

a) *BUILD (Resources)*

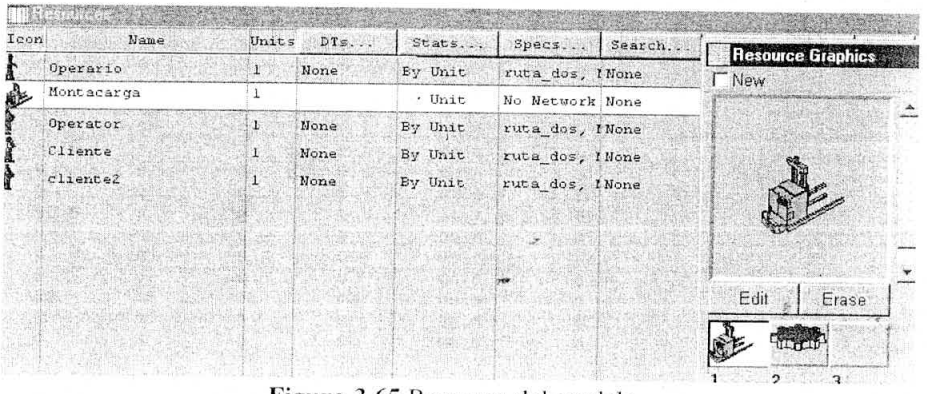
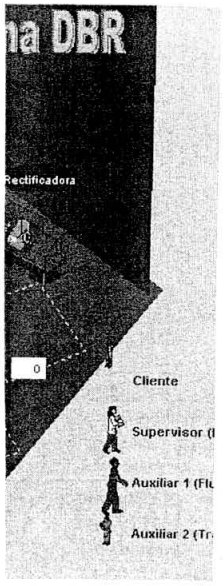


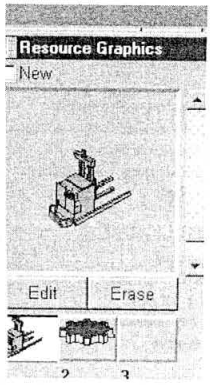
Figura 3.65 Recursos del modelo.

ceso son menores



:B.

ns, entities, path
os cambios son



b) *BUILD (Arrivals)*

Arrivals

Entity...	Location...	Qty each...	First Time	Occurrences	Frequency	Logic	Disable
pedido	Consumidor	1	0	10	5		No
pieza_4	Proveedor	50	Wk 1, Wed @ 12:	1	1		No
pieza	zona_cuatro	2	0	1	1		Yes
pieza_3	Zona_dos	2	0	1	1		No
pieza_2	zona_tres	2	0	1	1		Yes

Figura 3.66 Llegadas al sistema.

c) *BUILD (Processing)*

Entity	**Location**	**Operation**	**Output**	**Destination**	**Rule**	**Move Logic**
pedido	Rec_troqueladora	Sub_wip()	pedido	Pedidos_troqueladora	First 1	
pedido	Pedidos_troqueladora	IF (CONTENTS(ZonaMat_cuatro) <=2) THEN { ORDER 1 pedido TO Oc_Troqueladora } sub_wip()	pedido	ZonaMat_cuatro	JOIN 1	
pieza	ZonaMat_cuatro	sub_wip() JOIN 1 pedido	pieza_2	Troqueladora	First 1	MOVE WITH Operario THEN FREE
pieza_2	Troqueladora	sub_wip() WAIT 5 min	pieza_2	ZonaEmbarque	First 1	ct = CLOCK (MIN) - temp MOVE WITH Operario THEN FREE
pieza_2	ZonaEmbarque	sub_wip()	pieza_2	Producto_Terminado	First 1	MOVE WITH cliente2 THEN FREE
pieza_2	Producto_Terminado	sub_wip() WAIT 5hr	pieza_2	EXIT	First 1	
pedido	Oc_troqueladora	sub_wip()	pedido	**Recepción**	First 1	MOVE WITH Operator THEN FREE
pedido	ZonaPT_tres	sub_wip() WAIT 5hr	pedido	EXIT	First 1	
pieza_2	ZonaMat_tres	sub_wip()	pieza	Rectificadora	First 1	MOVE WITH Operario THEN FREE
pieza	Rectificadora	sub_wip()	pieza	ZonaMat_cuatro	First 1	MOVE WITH Operario THEN FREE
pedido	Pulidora	sub_wip()	pedido	EXIT	First 1	

Entity	Location	Operation	Output	Destination	Rule	Move Logic
pieza_3	ZonaMat-dos		pieza_2	Pulidora	First 1	MOVE WITH Operario THEN FREE
pieza_2	Pulidora	sub_wip() WAIT 3 min	pieza_2	ZonaMat_tres	First 1	MOVE WITH Operario THEN FREE
pedido	Recepción	sub_wip()	pedido	Proveedor	JOIN 1	
pieza_4	Proveedor	JOIN 1 pedido temp=CLOCK(MIN)	pieza_3	ZonaMat-dos	First 1	MOVE WITH Operario THEN FREE
pedido	Consumidor		pedido	Rec_troqueladora	First 1	MOVE WITH Cliente THEN FREE

Como se observa, los cambios con respecto al proceso del modelo **CONWIP** son mínimos.

Análisis de resultados

VARIABLES

Variable Name	Total Changes	Average Minutes Per Change	Minimum Value	Maximum Value	Current Value	Average Value
ct	10	14.75	1	109.31	109.31	107.92
th	118	1.39	0	7	0.01	0.04
qzona cuatro	0	0.00	0	0	0	0
ordenes	0	0.00	0	0	0	0
ordenes troqueladora	0	0.00	0	0	0	0
qtroqueladora	0	0.00	0	0	0	0
ordenes rectificadora	0	0.00	0	0	0	0
qrectificadora	0	0.00	5	5	5	5
ordenes pulidora	0	0.00	0	0	0	0
wip2	0	0.00	0	0	0	0
wip	118	1.39	0	10	2	2.10
wip3	0	0.00	0	0	0	0
wip4	0	0.00	0	0	0	0
CT orden	0	0.00	0	0	0	0
q conwip	0	0.00	50	50	50	50
th conwip	0	0.00	0	0	0	0
hora	0	0.00	0	0	0	0

Figura 3.67 Resultados del modelo DBR.

Parámetro	CT	TH	WIP
MRP	1418	0.2	96
JIT	125	7	10
CONWIP	87	3	10
DBR	125	7	10

Tabla 3.3 Comparación de los resultados entre los prototipos *push* MRP y pull CONWIP, DRB y JIT.

ɔn	Rule	Move Logic
	First 1	MOVE WITH Operario THEN FREE
ɪr	First 1	MOVE WITH Operario THEN FREE
	JOIN 1	
	First 1	MOVE WITH Operario THEN FREE
ɛl	First 1	MOVE WITH Cliente THEN FREE

ɔroceso del modelo

mum lue	Current Value	Average Value
.31	109.31	107.92
7	0.01	0.04
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
5	5	5
0	0	0
0	0	0
10	2	2.10
0	0	0
0	0	0
0	0	0
50	50	50
0	0	0
0	0	0

WIP
96
10
10
10

s *push* MRP y pull

Con esta información, Juan Carlos estaba en capacidad de recomendar a su gerente un sistema de producción *pull*, porque *DBR, CONWIP y JIT* funcionan mucho mejor que el *MRP* o sistema *push*. Pero ¿cuál de los sistemas *pull*? Definitivamente el *JIT o KANBAN* se puede descartar por el excesivo control sobre los inventarios. La respuesta a la pregunta si el *CONWIP o el DBR* es mejor dependerá de la complejidad del sistema de producción y del nivel de almacenamiento de los *buffers* o *inventarios de seguridad*. De acuerdo con la simulación, el *DBR* tiene mejor comportamiento.

4

Casos de logística

Caso 1. Cargue y descargue de camiones

ServiECI es una empresa de servicios dedicada a la recolección y entrega de mercancías en las principales ciudades de Colombia. En Bogotá, la empresa tiene una bodega central a donde llegan, por una parte, grandes tractomulas y camiones, llamados genéricamente *vanes*, que normalmente viajan de noche, y cubren la geografía nacional; y por otra parte, camioncitos pequeños, conocidos como *furgones*, que recorren la ciudad durante el día, recogiendo y entregando paquetes en las oficinas de la empresa, situadas en las distintas localidades o zonas en las que se encuentra dividida la capital.

A Darío Saavedra, ingeniero industrial encargado de mejorar los procesos de ServiECI, se le encomendó la tarea de analizar cómo se cargan y descargan los *vanes* y proponer alternativas de mejoramiento del proceso.

Actualmente, tan pronto llega un *van* a la bodega, se asigna un equipo de siete personas para descargarlo y volverlo a cargar. También se asigna una zona demarcada en el piso, de la llamada zona de embarque, para realizar estas operaciones. Tanto para cargue como para descargue, dos operarios se suben al camión y van colocando o alcanzando los paquetes, mientras que los demás alcanzan o reciben los paquetes, que trasladan desde o hasta la zona de embarque, subdividida en bloques demarcados en el piso, según las ciudades de destino. En el proceso actual, cada paquete se toma y alcanza individualmente.

Darío había aprendido en sus clases de producción que uno de los secretos del éxito de los japoneses, con su Sistema Justo A Tiempo, había sido la reducción al máximo de los tiempos de alistamiento (*Set up times*), separando los tiempos de operación, de los tiempos de preparación. El in-

geniero Saavedra pensó que al utilizar estibas (*pallets*), el equipo podría "adelantar trabajo", agrupando paquetes con el mismo destino y colocándolos en una estiba, antes o después de llegar el *van*, porque en el proceso actual, si no hay *van* para cargar o descargar, los operarios se sientan a esperar hasta que llegue un vehículo. Esto equivalía a encontrar y separar un tiempo de preparación del tiempo real de operación.

Sin embargo, el uso de estibas requería, adicionalmente, un montacargas, debido al peso de la estiba cargada. Pero un montacargas puede dejar o tomar la estiba sólo en la puerta del *van*, de manera que los operarios tendrían que "desempacar" manualmente la estiba y colocar los paquetes en el fondo del *van*, en el momento del cargue, teniendo que hacer la operación de manera inversa en el momento del descargue. Uno de los operarios le sugirió al ingeniero Saavedra el uso de una banda portátil o fija dentro del *van* para correr las estibas hasta o desde el fondo, porque él había visto este sistema en los compartimentos de carga de los aviones, cuando trabajaba en el aeropuerto.

A medida que pensaba en posibles soluciones, también iban apareciendo nuevas restricciones o problemas: un solo montacargas posiblemente no alcanzaría a cubrir varios *van* que llegasen simultáneamente; utilizar otra banda portátil para "subir"y "bajar" las estibas, con "traslado" desde la puerta hasta el fondo del *van* mediante otra banda, remplazaría un montacargas; también podría encargar una banda portátil que transportara las estibas desde el suelo hasta el fondo del *van*.

Para Darío, pensar en alternativas de mejoramiento era lo más sencillo. Lo verdaderamente complicado estaba en presentar y convencer al gerente Guerrero de la factibilidad de cualquiera de sus alternativas, porque él, como la gran mayoría de los gerentes, entendía únicamente en lenguaje monetario y de cifras, especialmente el que mostrara una ostensible reducción de costos. En esta situación, al ingeniero Saavedra no le quedaba otro camino que "probar" sus alternativas y cuantificarlas mediante simulación. Afortunadamente, durante su carrera había escogido las materias electivas que tenían relación con esta poderosa herramienta de análisis.

Así las cosas, el ingeniero planteó tres alternativas:

- En la primera, que llamó sistema actual, los operarios cargan y descargan, paquete por paquete, tan pronto llega el *van*.

- En la segunda, que llamó sistema con estibas o *pallets*, los operarios clasifican y colocan los paquetes en las estibas antes de llegar el *van*, y luego, con un montacargas, colocan o retiran las estibas de la puerta del *van*.
- En la tercera, que denominó sistema de banda, utiliza estibas, montacargas y bandas para el cargue y descargue.

Modelando el sistema de cargue y descargue de camiones

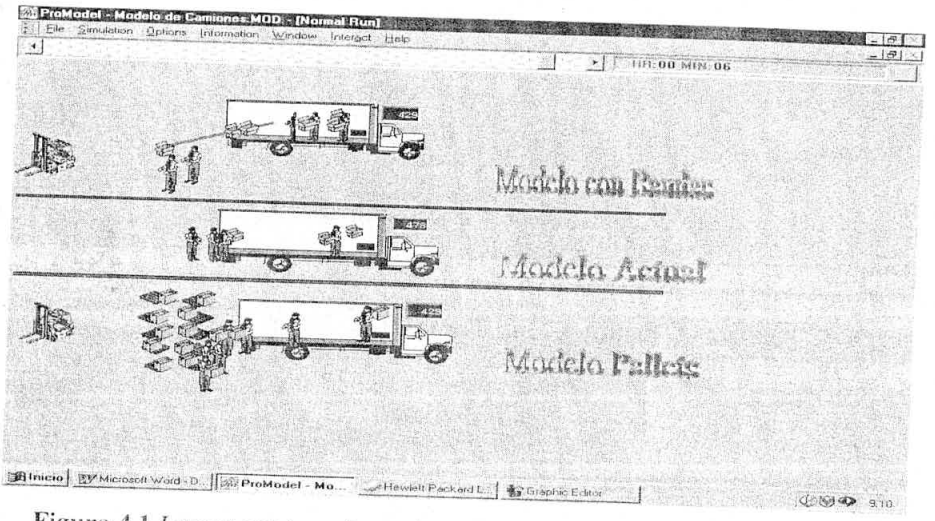


Figura 4.1 *Layout* con tres alternativas para cargue y descargue de camiones.

El modelo de simulación que preparó Darío tenía las siguientes partes:

1. *BUILD (Locations)*

Locations [21]

Icon	Name	Cap.	Units	DTs...	Stats...	Rules...	Notes...
	Banda	inf	1	None	Time Series	Oldest	
	Loc2	INFINIT	1	None	None	Oldest, FIFO	
	Loc3	inf	1	None	None	Oldest	
	Normal	inf	1	None	Time Series	Oldest	
	Loc5	inf	1	None	None	Oldest	
	Loc6	inf	1	None	None	Oldest	
	Loc7	inf	1	None	None	Oldest	
	Loc8	inf	1	None	None	Oldest	
	Pallets	inf	1	None	Time Series	Oldest	
	Loc10	inf	1	None	None	Oldest	
	pallet	INF	1	None	None	Oldest	
	pallet2	INF	1	None	None	Oldest	
	pallet3	INF	1	None	None	Oldest	
	pallet4	INF	1	None	None	Oldest	
	pallet5	INF	1	None	None	Oldest	
	pallet6	INF	1	None	None	Oldest	
	pallet7	INF	1	None	None	Oldest	
	pallet8	INF	1	None	None	Oldest	
	pallet9	INF	1	None	None	Oldest	
	pallet10	INF	1	None	None	Oldest	
	Loc11	INF	1				

Figura 4.2 Instalaciones del modelo.

2. *BUILD (Entities)*

Entities [5]

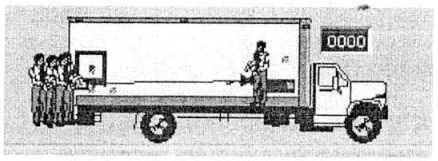
Icon	Name	Speed (mpm)	Stats...	Notes...
	caja	50	Time Series	
	Pallet_lleno	50	Time Series	
	caja2	50	Time Series	
	BOX	50	Time Series	
	PALLET_CARGADO	50	Time Series	

Figura 4.3 Entidades del modelo.

3. *BUILD (Path Network)*

Path Networks [1]

Graphic	Name	Type	T/S	Paths...	Interfaces...	Mapping...	Nodes
	Net1	Passing	Speed & Dista	1	2	0	2
	net2	Passing	Speed & Dista	2	3	0	3



Interfaces

Node	Location
N1	Normal
N2	Loc5

Figura 4.4 *Red 1* del modelo.

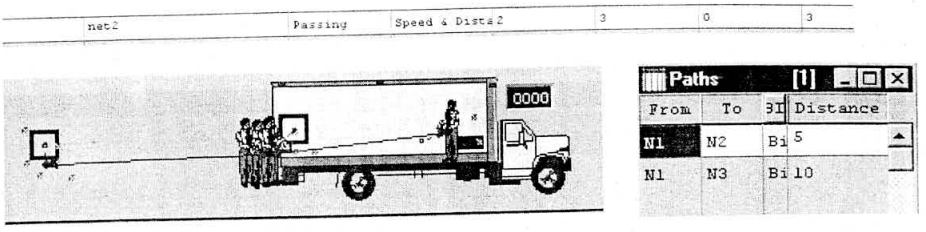


Figura 4.5 *Red 2* del modelo.

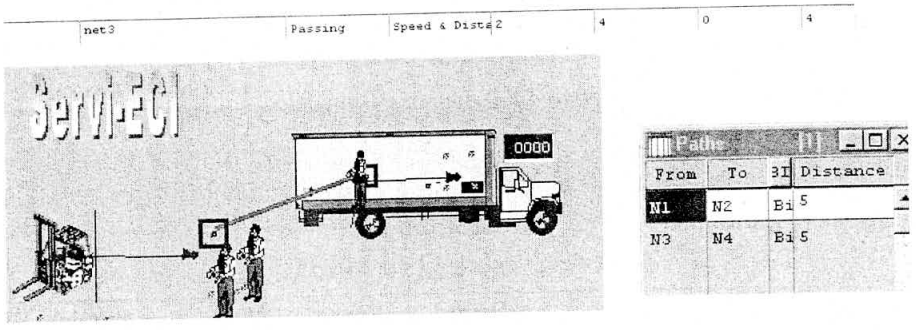


Figura 4.6 *Red 3* del modelo.

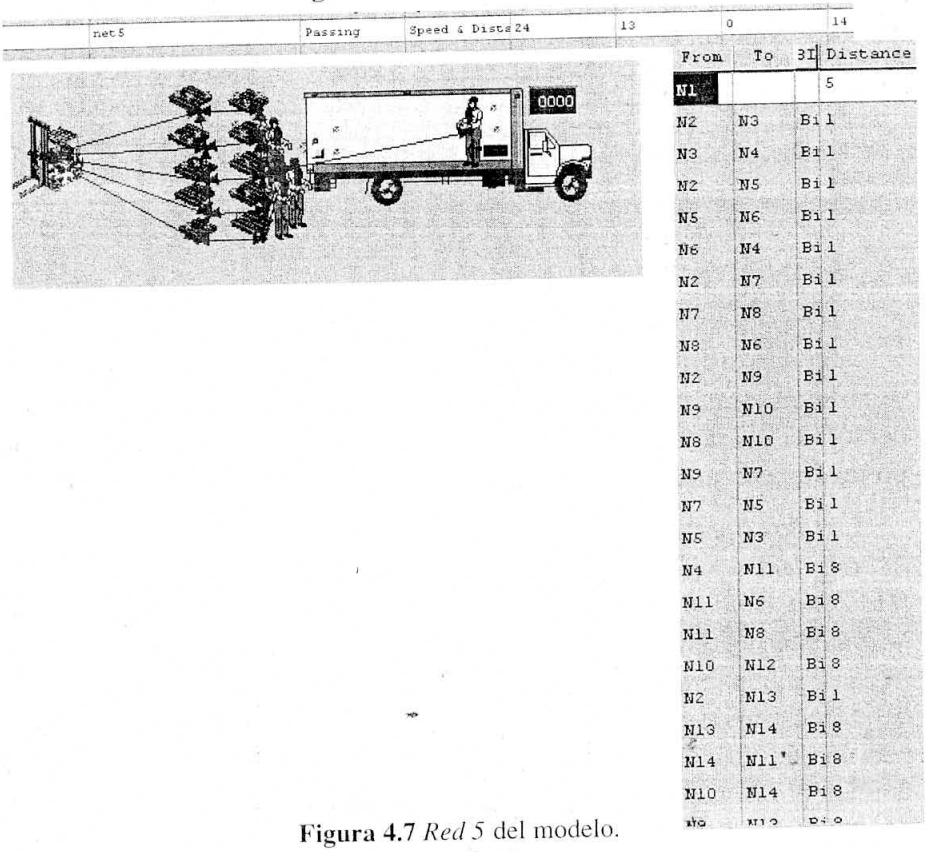


Figura 4.7 *Red 5* del modelo.

4. *BUILD (Resources)*

Resources

Icon	Name	Units	DTs...	Stats...	Specs...	Search...	Logic...	Pts...
	Banda_dentro_veh	4	None	By Unit	net3, N2,	None	0	1
	Banda_fuera_veh	2	None	By Unit	net3, N3,	None	0	2
	Actual_dentro_veh	2	None	By Unit	net2, N3,	None	0	1
	Actual_fuera_veh	5	None	By Unit	net2, N1,	None	0	4
	Pallet_dentro_veh	4	None	By Unit	net5, N1,	None	0	1
	Pallet_fuera_veh	3	None	By Unit	net5, N2,	None	0	5
	Montacarga2	1	None	By Unit	net5, N12	None	0	1
	Montacarga	1			net3, N4,	None	0	1

Figura 4.8 Recursos del modelo.

Los recursos los clasificó en cuatro grupos: bandas (6), operarios (7), estibas (7), montacargas (2) y los definió así:

Name	Units	Net	Home Node	Empty mpm	Full mpm	Pickup seconds	Deposit seconds
Banda_dentro_veh	4	Net 3	N2	n(40,5)	n(40,5)	n(3,1)	n(3,1)
Banda_fuera_veh	2	Net 3	N3	n(40,5)	n(40,5)	n(3,1)	n(3,1)
Actual_dentro_veh	2	Net 2	N3	n(40,5)	n(40,5)	n(3,1)	n(3,1)
Actual_fuera_veh	5	Net 2	N1	n(40,5)	n(40,5)	n(3,1)	n(3,1)
Pallet_dentro_veh	4	Net 4	N1	n(40,5)	n(40,5)	n(3,1)	n(3,1)
Pallet_fuera_veh	3	Net 4	N2	n(40,5)	n(40,5)	n(3,1)	n(3,1)
Montacarga2	1	Net 4	N12	50	50		
Montacarga	1	Net 3	N4	50	50		

Como se puede observar dentro del vehículo, las cuatro bandas se mueven con una distribución normal de 40 metros por minuto en promedio y desviación estándar de 5 mpm, tanto llenas como vacías; el tiempo que demora el operario en recoger o depositar en la banda es 3 segundos en promedio, con una desviación de 1 segundo.

Los dos operarios que trabajan dentro del *van* están definidos como *Actual_dentro_veh* y se mueven con la trayectoria *Net2*, que tiene tres nodos: *N1=Zona_camión_actual, N2=Descargue_actual y N3=Camión_actual.* Cuando los operarios se encuentran libres, permanecen en el nodo 2, clasificando o preparando las estibas. Como se dijo, estos operarios recogen y depositan paquetes a una velocidad normal de 3 segundos en promedio, con una desviación estándar de 1 segundo. Los otros cinco operarios se definen por *Actual_fuera_veh*, se mueven dentro de la trayectoria *Net2* y permanecen en el nodo 1 cuando están libres.

Uno de los montacargas se destinó a los *pallets* mediante la trayectoria *Net4*, el otro debe permanecer en la trayectoria *Net3*, comentada anteriormente. Estos recursos se movilizan vacíos o cargados a una velocidad de 50 metros por minuto.

Al sistema de cargue y descargue llegan tres entidades: *Caja, Caja2 y Box*, definidas en los *arrivals*, de la siguiente manera:

5. *BUILD (Arrivals)*

Arrivals [4]

Entity...	Location...	Qty each...	First Time	Occurrences	Frequency	Logic	Disabl
caja	Banda	unidades	0	1	1		No
caja2	Normal	unidades	0	1	1		No
BOX	Pallets	unidades	0	1	1		No

Figura 4.9 Llegadas al sistema.

Como el lector podrá apreciar, cada tipo de unidad corresponde a una alternativa. Todas las entidades llegan al sistema, cada minuto.

6. *BUILD (Variables)*

Variables (global)

Icon	ID	Type...	Initial value	Stats...	Not
No	num	Integer	0	Time Series	
No	tiempo	Real	0	Time Series	
No	temp_uno	Real	0	Time Series	
No	t1	Integer	0	Time Series	
No	t2	Integer	0	Time Series	
No	t3	Integer	0	Time Series	
No	unidades	Integer	0	Time Series	

Figura 4.10 Variables del modelo.

7. *BUILD (Attributes)*

Attributes

ID	Type...	Classification...
tiempo_caja	Real	Ent
tiempo_caja2	Real	Ent
tiempo_pallet	Real	Ent

Figura 4.11 Atributos del modelo.

8. *BUILD (Subroutines)*

Se definió una subrutina *mm* que tiene como fin desplegar, durante la simulación, un letrero que indica que el descargue se ha cumplido cuando el contenido simultáneo de las dos instalaciones en mención es cero.

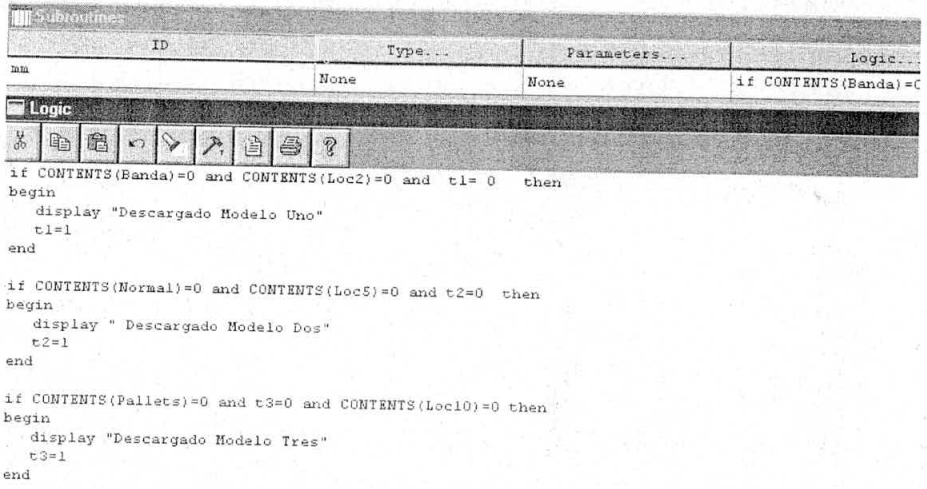


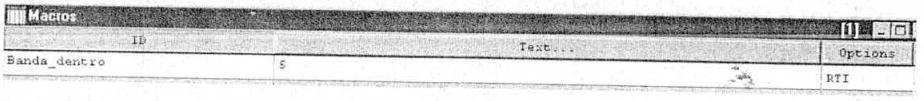
Figura 4.12 Subrutina mm del modelo.

9. *BUILD (General Information)*

En la *lógica de iniciación* que aparece en el menú *Build-General Information*, Darío colocó en el sitio *PROMT: "Ingrese el número de unidades"*, instrucción que permite cambiar el número de unidades al iniciar la simulación.

10. *BUILD (Macros)*

Darío definió un *macro* con el nombre de *Banda_dentro*, que le permite cambiar el número de bandas dentro de los *van*, e inició el campo *Texto* con un valor de 5.



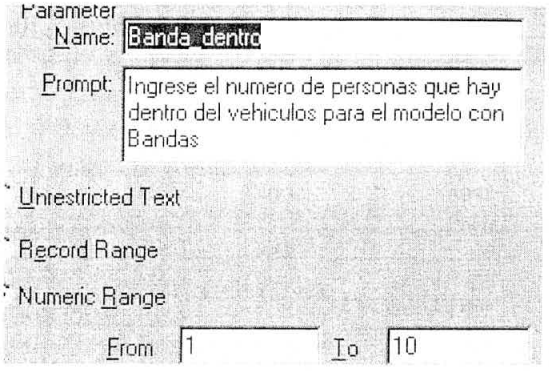


Figura 4.13 Macro para definir el número de personas dentro del *van*.

11. *BUILD (Processing)*

Como el proceso o *Processing* de este modelo es bastante prolongado, se irá comentando en grupos de cuatro instrucciones.

Entity	Location	Operation	Output	Destination	Rule	Move Logic
All	banda	tiempo_caja=clock() mm()	All	loc7	First 1	MOVE WITH Banda_dentro_veh THEN FREE
All	loc7	mm()	All	loc2	First 1	
All	loc2	mm()	All	loc3	First 1	
caja2	Normal	tiempo_caja2 =clock()	caja2	loc5	First 1	MOVE WITH Actual_dentro_veh THEN FREE

En la primera instrucción de este grupo, la entidad correspondiente, en este caso *Caja*, llega a la instalación *banda*, se guarda la hora del reloj de simulación en la variable *Tiempo_caja*; si ya se ha descargado, debe aparecer el aviso (ver subrutina mm). De manera similar, cundo la entidad *caja2* llega a la instalación *Normal* se guarda el tiempo en la variable *Tiempo_caja2*.

Entity	Location	Operation	Output	Destination	Rule	Move Logic
caja2	loc5	mm()	caja2	loc6	First 1	MOVE WITH Actual_fuera_veh or Actual_dentro_veh THEN FREE
caja2	loc6	mm()	caja2	EXIT	First 1	log "Tiempo de descargue2",tiempo_caja2
caja	banda		caja	EXIT		
caja	loc3	mm() GROUP n(30,5) AS Pallet_lleno	caja	EXIT		

En la tercera instrucción de este grupo, 30 *Cajas* se convierten en un *Pallet_lleno* mediante el comando *GROUP*. Obsérvese que en el *Output* de esta instrucción, la entidad vieja sale del sistema.

Entity	Location	Operation	Output	Destination	Rule	Move Logic
pallet_lleno	loc3	inc num mm()	pallet_lleno	loc8		MOVE WITH Montacarga THEN FREE
pallet_lleno	loc8	mm() USE Banda_fuera_veh FOR .5min	pallet_lleno	EXIT	First 1	log "Tiempo de descargue_banda1",tiempo_caja
box	pallets	tiempo_pallet=clock()	box	loc10	First 1	MOVE WITH Pallet_fuera_veh THEN FREE
box	loc10	mm()	box box box box box box box box box	pallet2 pallet pallet3 pallet4 pallet5 pallet6 pallet7 pallet8 pallet9 pallet10	Turn1 Turn Turn Turn Turn Turn Turn Turn Turn Turn	MOVE WITH Pallet_fuera_veh THEN FREE (Esta intrucción es igual para todos)

En la segunda instrucción de este grupo, se encuentra el comando *USE*, que tiene el efecto de utilizar el recurso, en este caso, *Banda_fuera_veh* durante 0.5 minutos. Además, en la lógica aparece el comando *LOG*, que escribe bajo el título *Tiempo de descargue_banda1* el contenido del atributo *Tiempo_caja*, donde se cronometra el tiempo de llegada de las *cajas*.

En la cuarta instrucción del grupo, en la *lógica de movimiento*, para cada pareja *Box, pallet i*, aparece *Turn*, que indica que el recurso debe hacer el movimiento indicado por riguroso turno. Mediante *MOVE WITH Pallet_fuera_veh THEN FREE*, el recurso disponible va colocando la entidad *Box* en los 10 *Pallets* disponibles.

Entity	Location	Operation	Output	Destination	Rule	Move Logic
box	pallet	GROUP 10 AS Pallet_cargado	box	EXIT		
box	pallet2	GROUP 10 AS Pallet_cargado	box	EXIT		
box	Pallet3	GROUP 10 AS Pallet_cargado	box	EXIT		
box	Pallet4	GROUP 10 AS Pallet_cargado	box	EXIT		
box	Pallet5	GROUP 10 AS Pallet_cargado	box	EXIT		
box	Pallet6	GROUP 10 AS Pallet_cargado	box	EXIT		
box	Pallet7	GROUP 10 AS Pallet_cargado	box	EXIT		
box	Pallet8	GROUP 10 AS Pallet_cargado	box	EXIT		
box	Pallet9	GROUP 10 AS Pallet_cargado	box	EXIT		
box	Pallet10	GROUP 10 AS Pallet_cargado	box	EXIT		

En este grupo de instrucciones, la entidad "box" se agrupa de a diez y se colocan en un *pallet* conformando una nueva entidad llamada *Pallet_cargado*. Le entidad vieja sale del sistema.

Entity	Location	Operation	Output	Destination	Rule	Move Logic
Pallet_cargado	pallet	mm()	Pallet_cargado	loc11	First 1	MOVE WITH Montacarga2 THEN FREE
Pallet_cargado	pallet2	mm()	Pallet_cargado	loc11	First 1	MOVE WITH Montacarga2 THEN FREE
Pallet_cargado	pallet3	mm()	Pallet_cargado	loc11	First 1	MOVE WITH Montacarga2 THEN FREE
Pallet_cargado	pallet4	mm()	Pallet_cargado	loc11	First 1	MOVE WITH Montacarga2 THEN FREE
Pallet_cargado	pallet5	mm()	Pallet_cargado	loc11	First 1	MOVE WITH Montacarga2 THEN FREE
Pallet_cargado	pallet6	mm()	Pallet_cargado	loc11	First 1	MOVE WITH Montacarga2 THEN FREE
Pallet_cargado	pallet7	mm()	Pallet_cargado	loc11	First 1	MOVE WITH Montacarga2 THEN FREE
Pallet_cargado	pallet8	mm()	Pallet_cargado	loc11	First 1	MOVE WITH Montacarga2 THEN FREE
Pallet_cargado	pallet9	mm()	Pallet_cargado	loc11	First 1	MOVE WITH Montacarga2 THEN FREE
Pallet_cargado	pallet10	mm()	Pallet_cargado	loc11	First 1	MOVE WITH Montacarga2 THEN FREE
Pallet_cargado	loc11	mm() USE Pallet-fuera-veh FOR 5 MIN	Pallet_cargado	EXIT	First 1	LOG "tiempo de descargue con pallet", tiempo_pallet

En este grupo de instrucciones la entidad *Pallet_cargado*, en cada una de las zonas de *pallets*, es transportada por el recurso *Montacarga2*. En la instrucción final se utiliza el recurso *Pallet_fuera_veh* durante cinco minutos y se escribe el contenido del atributo *Tiempo_pallet*.

12. *SIMULATION (Options)*

Entre las opciones del modelo se escogió *time only*. No se colocó tiempo de corrida porque al iniciar la simulación se abre una ventana donde se coloca el número de unidades con las que está cargado el *van*. El modelo termina cuando se descarga y carga el *van*.

Análisis de los resultados

Darío corrió la simulación para 500 unidades. Los resultados aparecen en la figura 4.14. Los operarios del sistema actual demoran 150 minutos aproximadamente para descargar un *van*. Si se utilizan estibas y montacarga, el tiempo se reduce casi a la tercera parte, esto es, 50 minutos aproximadamente. Sin embargo, la mejor alternativa es utilizar estibas y bandas transportadoras simultáneamente, que reduce el tiempo de descargue a 40 minutos. Adicionalmente, como se observa en la figura 4.15, el número de operarios se puede reducir a seis.

Aunque el software ProModel® permite incluir los costos en los recursos, Darío no los incluyó en este modelo, de manera que él debe complementar su análisis para poder dar una repuesta al gerente Guerrero. De todas maneras, con un gran volumen de unidades por descargar, se justifica la alternativa de introducir las bandas y los montacargas para reducir el tiempo de ciclo y, muy seguramente, aumentar la utilidad.

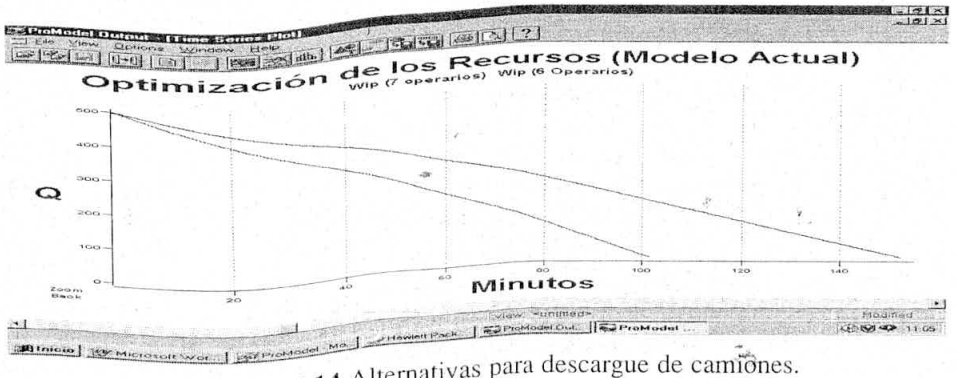


Figura 4.14 Alternativas para descargue de camiones.

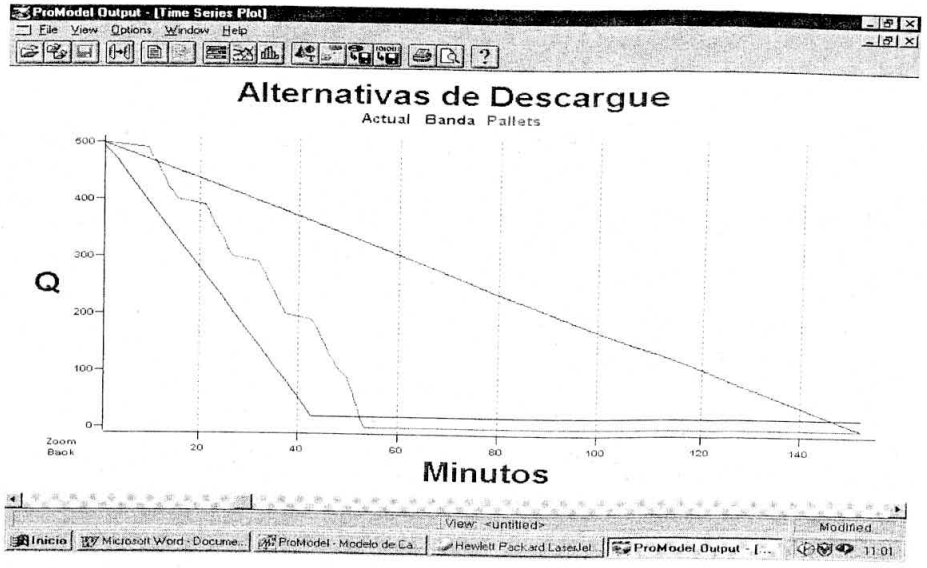


Figura 4.15 Alternativas para descargue de camiones.

Caso 2. Búsqueda y asignación de rutas

Otro de los problemas que a diario enfrentaba el ingeniero Darío Saavedra era la búsqueda de rutas, para que los *furgones* recorrieran la mínima distancia posible y, por tanto, gastaran el menor tiempo. ¿Cuál podría ser la ruta "óptima" para ir de un sitio a otro? era la pregunta del millón para Darío en ese momento.

Darío comenzó a recordar sus cursos de geometría y el sistema de coordenadas cartesianas, en el que la mínima distancia entre los puntos (x, y) y (a, b) está definida por la fórmula $\boldsymbol{d} = [(x - a)^2 + (y - b)^2]^{1/2}$, que aún no había olvidado. Pero muy pronto se dio cuenta de que cuando se quiere ir de un lugar a otro en *furgón*, las coordenadas cartesianas no tienen mucho sentido, porque los vehículos no viajan siempre en línea recta.

En uno de los cursos de su carrera de ingeniería industrial, el de distribución de plantas, más exactamente, Darío había oído hablar de las coordenadas rectilíneas, especiales para ir de un sitio a otro en sistemas de "manzanas" cuadradas, similares a los cuadernos "cuadriculados". Entonces buscó en su biblioteca el libro de Richard Francis[1], donde encontró la fórmula de la "distancia rectilínea" entre los puntos de coordenadas (x, y) y (a, b), definida así: $\boldsymbol{dr} = |x - a| + |y - b|$, donde las barras verticales corresponden a la función "valor absoluto". Pero ¿las manzanas de Bogotá sí son cuadradas? Fácilmente llegó a la conclusión que no lo son y que utilizar este método podía ser mejor que usar el cartesiano, aunque el error al emplearlo sería muy alto.

En sus cursos de investigación de operaciones había visto el método de Dijkstra para encontrar la ruta más corta en una red, pero lo único que

1 Richard Francis y John White, *Facility layout and location an analitical approach*, Nueva Jersey, Prentice-Hall International Series in Industrial and Systems Engineering, 1974, p. 169.

recordaba era su dificultad para aplicarlo. Alguien le dijo que en el Instituto Geográfico Agustín Codazzi vendían mapas a escala de Bogotá, de donde podría extraer la información de las distancias entre los diferentes sitios de la ciudad y de la forma de las manzanas. Sin embargo, ingresar dicha información al computador era "todo un camello".

Pensando en su problema, Darío recordó que en uno en los laboratorios de ingeniería civil de la Escuela Colombiana de Ingeniería, manejaban un software especializado *SIG (Sistema de Información Geográfica)* y que allí muy posiblemente encontraría versiones digitalizadas del mapa de Bogotá. No perdió el viaje hasta la Escuela, porque allí consiguió lo que buscaba: el mapa digital de Bogotá.

Sin embargo, Darío continuó pensando en las contravías, en las calles cerradas, en los famosos huecos, en los "trancones", en el pico y placa, en las prohibiciones a vehículos de carga, y muy pronto comprendió que su búsqueda debía ser de rutas "bastante buenas" y no de rutas de "óptimas", porque entre los factores que había que considerar, muchos no se podían controlar cuando se quería ir de un lugar a otro en *furgón*.

Modelando el sistema de asignación de rutas

Una vez que el ingeniero consiguió un mapa digitalizado de la ciudad, comenzó por "importarlo" a ProModel®. En estos mapas las distancias son las reales y basta con emplear la escala adecuada para obtenerlas. En la figura 4.16 se muestra una parte del mapa, que corresponde a la zona norte de Bogotá.

En el siguiente paso, estudió cuidadosamente el mapa, dio un paseo por la ciudad para comparar el mapa con la realidad, y así determinar el sentido de las vías, los cruces prohibidos, las calles cerradas, las vías en construcción, etcétera.

Con paciencia infinita, el ingeniero Saavedra fue definiendo lo que él llamó *malla vial*, que no es otra cosa que la especificación de la totalidad de trayectorias factibles, entre un conjunto de nodos, con el sentido de la vía incluido y con las distancias reales. Los nodos correspondían a las intersecciones de las vías y se definieron como *locations;* así por ejemplo, el nodo 1, *N1,* estaba ubicado en la *location: ave 127_cra7*, el nodo 2, *N2,* correspondía a la *location: ave 116_cra11*. Como se observa en la figura 4.16, la distancia entre los dos sitios es aproximadamente 1.049 metros, con sentido bidireccional.

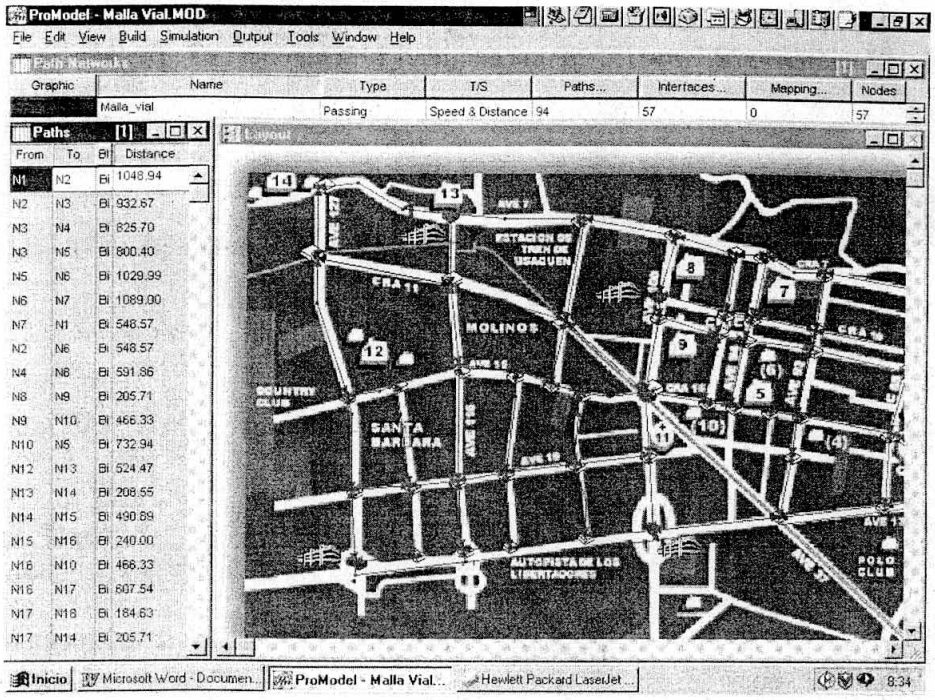


Figura 4.16 Mapa digitalizado de la zona nororiental de Bogotá, con los nodos que se definen en las intersecciones.

El gerente Guerrero pasaba por el puesto de trabajo del ingeniero Saavedra, a observar su trabajo, y se sorprendía al verlo "jugando" a los carritos para definir 60 *locations* o intersecciones, 93 trayectorias o *paths*, 52 *interfaces* y la red "malla_vial".

A continuación se muestran unos pocos datos, a manera de ejemplo, porque como el lector comprenderá, presentar la totalidad es muy largo y tedioso.

Cinco de las sesenta *locations*, son las siguientes:

Name	Capacity	Units	Statistics
Ave127_cra7	1	1	Time Series
Ave100_cra7	1	1	Time Series
Ave109_ave13	1	1	Time Series
Ave93_cra15	1	1	Time Series
Cll85_ave13	1	1	Time Series

Se incluyó solamente una entidad llamada *Mercancía*, que llega a la instalación *Ave127_ave15*, en una cantidad 1, ocurrencias 1, frecuencia 1.

Cinco de las 52 *interfaces*, son las siguientes:

Net	Node	Location
Malla_vial	N1	Ave127_cra11
	N2	Ave116_cra11
	N3	Ave106_cra11
	N4	Ave100_cra15
	N7	Ave127_cra7 (home)

Darío incluyó solo un *resource* que llamó *Machinist*, y lo definió así:

Name	Units	Net	Home Node	Empty mpm	Full mpm	Pickup seconds	Deposit seconds
Machinist	1	Malla_vial	N19	400mpm	400mpm		

Cinco de las instrucciones que colocó en la ventana *logic* del recurso *Machinist* son:

Resource	Node	Entry Logic	Exit Logic
Machinist	N7	WRITELINE PLANILLA, "Ave127 con cra 7ª."	
	N1	WRITELINE PLANILLA, "Ave127 con cra 11"	
	N2	WRITELINE PLANILLA, "Ave116 con cra 11"	
	N6	WRITELINE PLANILLA, "Ave116 con cra 7ª."	
	N5	WRITELINE PLANILLA, "Ave106 con cra 7ª."	

El *processing* de este modelo, como muestra sólo un viaje redondo de *furgón*, es corto y contiene la siguiente información:

Entity	Location	Op.	Output	Destination	Rule	Move Logic
Mercancia	Ave127_ave15		Mercancia	Ave116_cra7	First 1	**MOVE WITH** Machinist **THEN FREE**
Mercancia	Ave116_cra7		Mercancia	Ave13_cll100	First 1	**MOVE WITH** Machinist **THEN FREE**
Mercancia	Ave13_cll100		Mercancia	Cra81_cra7	First 1	
Mercancia	Cra81_cra7		Mercancia	Ave100_cra10	First 1	**MOVE WITH** Machinist **THEN FREE**
Mercancia	Ave100_cra10		Mercancia	Ave127_ave13	First 1	
Mercancia	Ave127_ave13		Mercancia	Ave127_ave15	First 1	**MOVE WITH** Machinist **THEN FREE**
Mercancia	Ave127_ave15		Mercancia	EXIT	First 1	

Figura 4.18 Escogencia de la mejor ruta por el modelo.

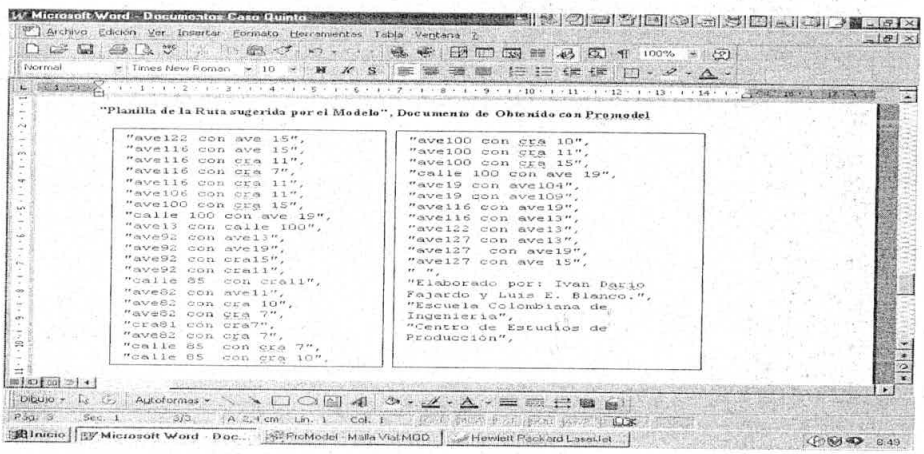


Figura 4.19 Direcciones de la mejor ruta sugerida por el modelo.

Caso 3. Cadena de abastecimiento

El gerente de operaciones de ServiECI, en una de las reuniones sociales del Club de Ingenieros, escuchó casualmente a dos ingenieros industriales, uno ya entrado en años, de apellido Camacho, y a una joven recién egresada llamada María Inés, que conversaban sobre el significado de la palabra *logística*.

Para el ingeniero Camacho la *logística* es el subsistema de la empresa que se encarga del manejo de materias primas, productos en proceso, productos terminados e inventarios; en ocasiones hace también mantenimiento a las máquinas y a elementos de transporte interno, como bandas, cintas transportadoras, elevadores, montacargas, estibas, entre otros.

Para María Inés, la *logística* es un sistema mucho más grande, que une a clientes, producción y proveedores, por medio de dos grandes flujos: uno de información y otro de valor agregado. El flujo de información parte del cliente en forma de pedidos, llega a producción, se verifica si existen inventarios y, si este es el caso, se devuelve al cliente en forma de flujo de valor agregado, de productos terminados. Si no hay inventarios, el flujo de información sigue hasta los proveedores, en forma de pedidos de materiales, y retorna nuevamente como flujo de valor agregado, de materiales, que más adelante se transforman en productos terminados.

La llegada de otras personas interrumpió la conversación, de manera que el gerente Guerrero no pudo conocer la conclusión del debate. El lunes siguiente llamó al ingeniero Darío Saavedra y le propuso que, "dada la importancia actual de la *logística*", hiciera una presentación breve, clara y concisa sobre el tema a ingenieros y supervisores una semana más tarde.

Darío pensó para sus adentros que el hecho de haber escogido *logística* dentro de las materias electivas técnicas, en su carrera de ingeniero indus-

trial, le brindaba ahora una nueva oportunidad para demostrar lo que sabía. Esa noche encontró en su biblioteca tres libros sobre el tema[2], uno de ellos muy reciente, y comenzó a preparar su intervención.

En *Logistic Engineering and Management* encontró la definición de *logística* del Consejo de Administración Logística[3]: "*Logística* es el proceso de planeación, implementación y control, eficiente y efectivo del flujo y almacenamiento de materias primas, inventarios de productos en proceso y productos terminados y de toda la información desde el punto de origen hasta el punto de consumo, con el propósito de satisfacer las necesidades de los clientes".

En el libro de publicación más reciente, *Introduction to Supply Chain Management*, sus autores hablan del renacimiento logístico actual, pero se centran en la hoy conocida *cadena de abastecimientos*, que definen de manera similar a la *logística*[4]: "La *cadena de abastecimientos* armoniza todas las actividades asociadas con el flujo y transformación de bienes desde la etapa de las materias primas (extracción) hasta el usuario final, así como todos sus flujos de información asociados. Los materiales y la información fluyen en los dos sentidos, hacia arriba y hacia abajo, dentro de la *cadena de abastecimientos*".

En *Logística y aprovisionamiento*, Christopher[5] define la *logística* como "el proceso de gestionar estratégicamente la obtención, movimiento, y almacenamiento de materias primas, componentes y existencias terminadas (y los flujos de información relacionada) a través de la organización y sus canales de *marketing* de tal forma que la rentabilidad futura se vea maximizada a través de la cumplimentación efectiva de los pedidos en relación con los costes".

Darío encontró que las tres definiciones eran muy similares. En ellas se destacan dos flujos: el de información y el de materiales en transformación o en proceso de agregación de valor. Por otra parte, existe toda una labor de gestión (dirección, planeación, programación, control) para que los pedidos de los clientes se satisfagan oportunamente y con calidad y precios

2 Martin Christopher, *Logística y aprovisionamiento. Cómo reducir costes, stocks y mejorar los servicios*, Biblioteca de Empresa, Barcelona, Ediciones Folio S.A., 1994.
Robert Handfield y Ernest Nichols, *Introduction to Supply Chain Management*, Nueva Jersey, Prentice Hall, 1999.
Benjamin Blanchard, *Logistics Engineering and Management*, 4th ed., Nueva Jersey, Prentice Hall, 1992.

3 *Ibid.*, p. 3.

4 Robert Handfield y Ernest Nichols, *op. cit.*, p. 2.

5 Martin Christopher, *op. cit.*, p. 12.

competitivos. La entendió como una fusión de servicio al cliente, compras, producción, despachos y distribución. Halló, además, que *logística* y *cadena de abastecimientos* son sinónimos.

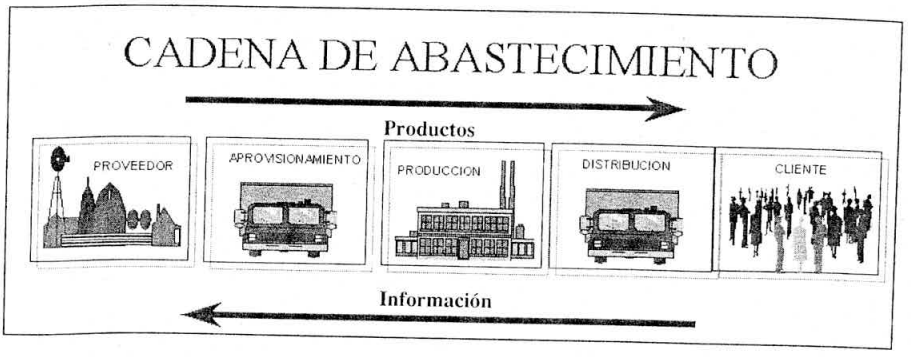


Figura 4.20 Partes que integran el sistema logístico o cadena de abastecimientos.

Después de su lectura, Darío comenzó a preparar carteleras, presentación Power Point® para su exposición, en donde se mostrarían gráficamente las actividades que se desarrollarían en cada una de las partes del sistema logístico y sus interrelaciones.

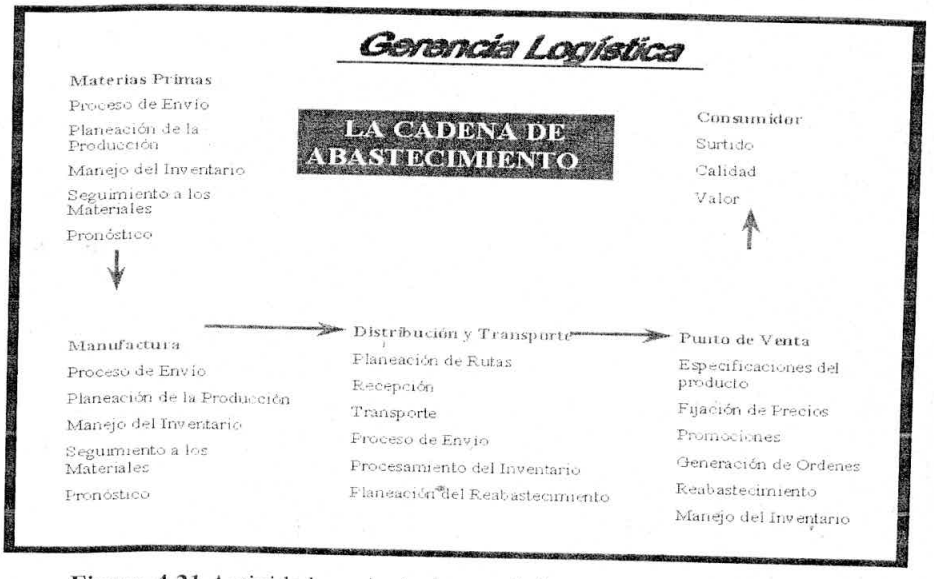


Figura 4.21 Actividades principales en el sistema logístico o cadena de abastecimientos.

Como se muestra en las figuras 4.21 y 4.22, siguiendo la dirección del flujo de valor agregado, las actividades logísticas comienzan con los proveedores, quienes actualmente se integran a los equipos de diseño de las empresas; se les hace verificación de sus sistemas de calidad, con el fin de evitar costosas y lentas inspecciones, y se establecen relaciones de exclusividad con ellos, una vez que garantizan calidad, cumplimiento y precios competitivos. La complejidad del envío de las materias primas depende de la ubicación de los proveedores, si son locales, nacionales o internacionales. En el caso de proveedores internacionales, es necesario preparar los documentos de nacionalización, seguros, transportes, bodegaje en puertos, etc. El sistema logístico debe hacer el seguimiento de los pedidos para conocer el lugar, la fecha y las condiciones de arribo.

En los sistemas de operaciones que siguen la filosofía Justo A Tiempo, las empresas envían a sus proveedores copias de sus pronósticos de demanda y cálculos de las materias primas que necesitarán en el futuro, para que ellos puedan, a su vez, preparar su envío en las fechas, sitios y condiciones de calidad requeridas.

Posteriormente viene el proceso de manejo de los materiales dentro de las instalaciones de la empresa, comenzando por la verificación de los pedidos y el ingreso de los materiales al almacén, ubicándolos en el lugar y en la forma adecuados. Después, las materias primas se van sacando de los almacenes por medio de las órdenes de producción. En el caso de materiales que requieren condiciones de transporte especiales, montacargas, rieles, ductos, etc., el sistema logístico se encarga de hacerlo. Lo mismo sucede a los almacenamientos de productos en proceso y terminados.

Después vienen las actividades de despacho de los pedidos y la distribución a los clientes, cuya complejidad dependerá del tipo de productos (tamaño, perecederos, refrigerados, frágiles, etc.) y de la distancia a donde se encuentren los clientes (centros de distribución, bodegas, almacenes, vanes, furgones, motos, entre otros).

Finalmente están las actividades relacionadas con la entrega, verificación de la exactitud del contenido, la fecha, dirección y entrega de los pedidos, más todas las que tienen que ver con el servicio al cliente: encuestas, promociones, reclamos, etcétera.

El ingeniero Saavedra tenía el panorama gráfico para presentar en la reunión, pero él quería mostrar que no es únicamente la complejidad la característica esencial de la *logística*, sino la interrelación entre todos los

subsistemas, tanto en el flujo de información como en el flujo de valor agregado. Así muy pronto recordó que una de sus aficiones era la simulación y que, por lo tanto, debía "sacarle partido".

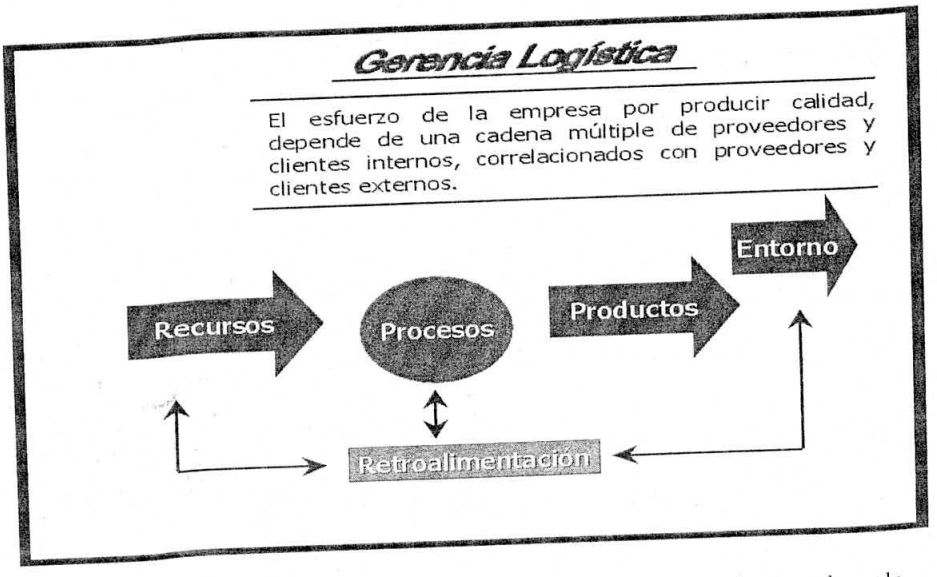


Figura 4.22 Papel integrador de la gerencia en el sistema logístico o cadena de abastecimientos.

Modelando el sistema logístico o cadena de abastecimientos

Mostrar la relación entre el flujo de información y el flujo de materiales de una manera más animada fue la meta principal que se trazó Darío para su modelo, además de cuantificar de alguna manera la satisfacción de los clientes. Pensó en una fábrica, que pide materiales a sus proveedores, mediante órdenes que se originan en un centro de cómputo y que responden a pedidos de los clientes. La fábrica envía productos terminados al centro de distribución de Cali, si es autorizada por el centro de cómputo; el centro de distribución de Cali, a su vez, envía productos terminados al centro de distribución de Bogotá, si es autorizado por el centro de cómputo y, finalmente, autorizado por el centro de cómputo, el centro de distribución de Bogotá envía productos al almacén, y éste los entrega al cliente. Los clientes que reciben su producto se contabilizan como clientes satisfechos.

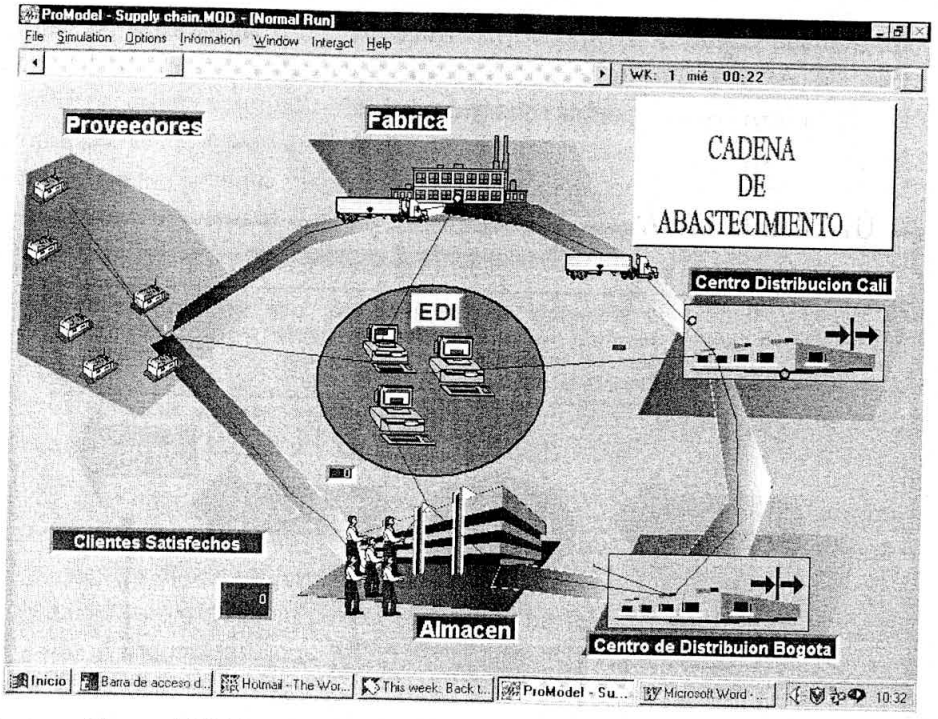


Figura 4.23 Modelo de simulación del sistema logístico o cadena de abastecimientos.

Para la simulación con ProModel®, Darío tuvo en cuenta 19 *locations* con esta información:

Name	Capacity	Units	Statistics
Fábrica	5	1	Time Series Oldest
Centro_Distribución_Cali	5	1	Time Series Oldest
Almacén_uno	5	1	Time Series Oldest
Zona_uno_fábrica	30	1	Time Series Oldest
Muelle_llegada_CD	30	1	Time Series Oldest
Muelle_llegada_CD_final	10	1	Time Series Oldest
Producto_terminado	30	1	Time Series Oldest
Tránsitos	30	1	Time Series Oldest
Despacho_fábrica	30	1	Time Series Oldest
Muelle_salida_CD	10	1	Time Series Oldest
Consumidor	Infinite	1	Time Series Oldest
Entrada_almacén	Infinite	1	Time Series Oldest
Centro_cómputo_uno	20	1	Time Series Oldest
Centro_cómputo_dos	20	1	Time Series Oldest
Centro_cómputo_tres	20	1	Time Series Oldest
Clientes_satisfechos	Infinite	1	Time Series Oldest
Proveedores	Infinite	1	Time Series Oldest
Almacén_dos	5	1	Time Series Oldest
Centro_Distribución_Bogotá	5	1	Time Series Oldest

Incluyó 7 *entities*:

Name	Speed	Statistics
Pieza_1	150	Time Series
Pieza_2	150	Time Series
Pieza_3	150	Time Series
Pieza_4	150	Time Series
Clientes	150	Time Series
Orden_uno	150	Time Series
Orden_dos	150	Time Series

Definió los *arrivals* de la siguiente manera:

Entity	Location	Qty. each	First Time	Occurrences	Frequency	Logic
Orden	Consumidor	N(2,1)	0	Infinite	e(15)	
Orden	Consumidor	20	Wk1, Wed @ 11 AM	1	1	

Darío definió una trayectoria con el nombre de *Ruta* con los siguientes *paths e interfaces*:

Path Networks	
From	To
N2	N3
N3	N4
N4	N5
N5	N6
N6	N7
N7	N8
N8	N9
N9	N10
N10	N11
N11	N12
N12	N13
N13	N14
N14	N15
N15	N2
N19	N12
N18	N7
N17	N4
N1	N2
N10	N20
N2	N17

Interfaces	
Node	Location
N1	Tránsitos
N2	Proveedores
N3	Zona_uno_fábrica
N4	Fábrica
N5	Depacho_fábrica
N6	Muelle_llegada_CD
N7	Centro_Distribución_Cali
N8	Muelle_salida_CD
N9	Muelle_llegada_CD_final
N10	Centro_Distribución_Bogotá
N11	Entrada_almacén
N12	Almacén_dos
N13	Producto_terminado
N14	Clientes_satisfechos
N15	Consumidor
N18	Centro_cómputo_dos
N19	Centro_cómputo_uno
N17	Centro_cómputo_tres
N20	Almacén_uno

Tuvo en cuenta un recurso llamado *Vehículo* con dos unidades, definido así:

Name	Units	Search	Path	Empty	Full	Home
Vehículo	2	Closest	Ruta	50 mpm	50 mpm	N1

Definió dos atributos reales para entidad con los nombres: *att* y *att_orden*, e incluyó además 14 variables con la siguiente información:

Identification	Type	Initial Value	Statistics
ct	Real	0	Time Series
th	Real	0	Time Series
ordenes_troqueladora	Integer	0	Time Series
qtroqueladora	Integer	0	Time Series
ordenes_rectificadora	Integer	0	Time Series
qrectificadora	Integer	5	Time Series
ordenes_pulidora	Integer	0	Time Series
wip2	Integer	0	Time Series
wip	Integer	0	Time Series
wip3	Integer	0	Time Series
wip4	Integer	0	Time Series
th	Integer	0	Time Series
ordenes	Integer	0	Time Series
ct_orden	Real	0	Time Series

Utilizó la subrutina *cal_th*, para calcular el *Throughput*, en la ventana *Logic*, mediante la ley de Little:

```
IF (ct <>0) THEN
 th=wip/ct
```

Las instrucciones del proceso se comentarán en grupos de cuatro:

Entity	Location	Operation	Output	Destination	Rule	Move Logic
All	Proveedores	**INC** wip att=**CLOCK**() cal_th()	All	Zona_uno_fábrica	First 1	**MOVE WITH** Vehículo **THEN FREE**
pieza_1	Zona_uno_fábrica	cal_th() **JOIN 1** orden_uno	pieza_2	Fábrica	First 1	**MOVE WITH** Vehículo **THEN FREE**
orden_uno	Centro_de_Cómputo_tres	cal_th()	orden_uno	Zona_uno_fábrica	**JOIN 1**	**MOVE FOR** 0.5
pieza_2	Fábrica	cal_th() **WAIT** e(10) **INC** wip3	pieza_2	Despacho_fábrica	First 1	**MOVE WITH** Vehículo **THEN FREE**

En la primera instrucción, cada vez que llegue una entidad a *Proveedores*, se incrementa la variable *Wip*, se guarda la hora de llegada en el atributo *att* y se calcula el *throughput* mediante la subrutina *cal_th()*. En las instrucciones segunda y tercera se observa el efecto del comando *JOIN 1* *orden_uno* sobre la entidad *pieza_1*, que ordena a la *pieza_1* que se junte con una *orden_uno* y a la entidad *orden_uno* que se junte con una entidad *pieza_1*. Obsérvese que el *JOIN* aparece en las dos instrucciones, una en *Operation* y la otra en *Rule*.

Entity	Location	Operation	Output	Destination	Rule	Move Logic
pieza_2	Despacho_ fábrica	cal_th()	pieza_2	Muelle_ Llegada_CD	First 1	**MOVE WITH** Vehículo **THEN FREE**
pieza_2	Muelle_ Llegada_CD	cal_th() **JOIN 1** orden_uno	pieza_3	Centro_ Distribución _Cali	First 1	**MOVE WITH** Vehículo **THEN FREE**
orden_uno	Centro_de_ Cómputo_dos	cal_th()	orden_u no	Muelle_ Llegada_CD	**JOIN 1**	**MOVE FOR** 0.5
pieza_3	Centro_ Distribución_ Cali	**DEC** wip3 **WAIT** n(20,10) **INC** wip2 cal_th()	pieza_3	Muelle_ Salida_CD	First 1	**MOVE WITH** Vehículo **THEN FREE**

Estas cuatro instrucciones son muy conocidas por los lectores, de manera que no se comentan.

Entity	Location	Operation	Output	Destination	Rule	Move Logic
pieza_3	Muelle_ Salida_CD	cal_th()	pieza_3	Muelle_ Llegada_CD _final	First 1	**MOVE WITH** Vehículo **THEN FREE**
pieza_3	Muelle_ Llegada_CD_ final	**JOIN 1** orden_uno cal_th()	pieza_4	Almacén	First 1	**MOVE WITH** Vehículo **THEN FREE**
orden_uno	Centro_de_ Cómputo_uno	cal_th()	orden_u no	Muelle_ Llegada_CD _final	**JOIN 1**	**MOVE FOR** 0.5
pieza_4	Almacén	**DEC** wip2 **WAIT** e(15) **INC** wip4 cal_th() ct=**CLOCK**()-att	pieza_4	Entrada_ almacén	First 1	**MOVE WITH** Vehículo **THEN FREE**

En este grupo, el último comando de la cuarta instrucción, calcula el *tiempo de ciclo ct* restándole a la hora actual, la hora de entrada que se había guardado en el *atributo att*. Las demás instrucciones son conocidas.

Entity	Location	Operation	Output	Destination	Rule	Move Logic
pieza_4	Entrada_ almacén	cal_th()	pieza_4	Producto_ terminado	**JOIN 1**	**MOVE WITH** Vehículo **THEN FREE**
orden_uno	Consumidor	Att_orden= **CLOCK()** cal_th()	Clientes	Producto_ terminado	First 1	**INT** cant cant=**CONTENTS**(Zona_uno_fábrica) **IF**(wip3<=5)**THEN** **begin** **ORDER** 10 orden_uno **TO** Centro_de_Cómputo_tres **IF** (cant<=5)**THEN** **ORDER** n(10,1) pieza_1 **TO** Proveedores **end** **IF**(wip2<=0)**THEN** **begin** **ORDER** 5 orden_uno **TO** Centro_de_Cómputo_dos **end** **IF** (wip4<=1)**THEN** **begin** **ORDER** 5 orden_uno **TO** Centro_de_Cómputo_uno **end** **MOVE WITH** Vehículo **THEN FREE**
Clientes	Producto_ terminado	**DEC** wip4 cal_th() **JOIN 1** pieza4 Ct_orden= **CLOCK()**-att_orden	Clientes	Clientes_ satisfechos	First 1	**MOVE FOR** 0.5
All	Clientes_ Satisfechos	**WAIT** 300hr	All	EXIT	First 1	MOVE FOR **0.5**

En la segunda instrucción de este grupo, en el *Move Logic* se encuentra una cadena de *IF* con la siguiente función. En las variables *wip3*, *wip2* y *wip4* se lleva la cuenta del número de *órdenes* que hay en los centros de cómputo *tres*, *dos* y *uno*, respectivamente. Mediante el comando *INT cant* se define una variable local llamada *cant* en donde se guarda el contenido de la *zona uno fábrica*, o sea el número de unidades que hay guardadas en la zona de almacenamiento de la fábrica. Cuando dicha cantidad es menor que 5, se ordenan 10 piezas a los proveedores. Si el número de *órdenes* en el *centro de cómputo tres* es menor que o igual a 3, se piden 10 más. De manera similar, se procede con las *órdenes* de los otros dos *centros de cómputo*.

En la tercera instrucción, mediante el comando *ct_orden=CLOCK()-att_orden*, se calcula el tiempo de ciclo de la orden, restándole a la hora actual el tiempo de llegada de la orden.

Presentando el sistema logístico o cadena de abastecimientos al gerente

Dentro del plazo convenido, con los gráficos, la presentación Power Point® y el sistema logístico en ProModel, el ingeniero Darío hizo su exposición ante el gerente Guerrero, los ingenieros y supervisores de ServiECI, con bastante éxito, porque todos entendieron con facilidad las interrrelaciones y la complejidad del sistema logístico en la versión actual.

Al gerente Guerrero le quedó muy claro que la idea de la ingeniera María Inés era la correcta. Felicitó al ingeniero Saavedra por la claridad de su presentación, especialmente por haber incluido en el modelo de simulación la idea de controlar el *Work In* Process, y encomendó al grupo la organización de todo el sistema logístico de ServiECI con los parámetros que Darío había mostrado en su exposición.

Caso 4. ¿Qué es eso del *crossdocking*?

Unión Logística de Colombia es una empresa dedicada al almacenamiento y transporte de bienes. Para la clasificación y almacenamiento de productos, la empresa posee una enorme bodega situada en la zona industrial de la ciudad. El principal accionista de la empresa es Manuel Osorio, quien transformó "a puro pulso" su pequeña empresa de transporte de carga en una grande de operación logística. Como la planeación y el control de las actividades logísticas cada día se hacían más complejas, Manuel contrató al ingeniero Mario Fajardo, experto en manejo de sistemas de información y en planeación y programación de operaciones.

La primera tarea que Manuel asignó al ingeniero Mario fue averiguar *qué era eso del crossdocking*. En la opinión de otro accionista, "eso" lo estaban aplicando en una empresa de la competencia con bastante éxito. Mario recurrió a sus apuntes, libros y no encontró nada, por ser éste un tema muy específico. Su amiga y compañera de estudios Sonia le sugirió que buscara en *Internet* porque en la universidad ella había buscado y encontrado algo sobre ese tema, pero ya no se acordaba bien. Mario hizo la exploración sugerida y esto fue lo que encontró:

> El *crossdocking* es una técnica logística relativamente nueva, utilizada en las industrias de reventa y transporte de carga, mediante la cual rápidamente se organizan los embarques provenientes de fuentes diversas y se realizan economías de escala en su asignación y transporte hacia el exterior. La técnica del *crossdocking* esencialmente elimina la función de mantenimiento de inventario de una bodega y más bien la utilizan para funciones de organización de embarques y envíos. La idea central del muelle de intercambio rápido (*crossdock)* consiste en transferir directamente de los embarques que llegan en vehículos de

carga a los embarques que salen en otros vehículos, sin almacenar los productos. Los embarques en promedio gastan en el muelle menos de 24 horas y a veces menos de una hora.

*¿Cuál es la diferencia en el funcionamiento de una bodega tradicional y una con muelle de intercambio rápido (*crossdock*)?*

En una bodega tradicional, los bienes se reciben de los proveedores y son almacenados en los sitios asignados con la ayuda de medios como estibas (*pallets*) o estantes (*shelves*). Cuando un cliente (ejemplo, el usuario o un almacén de reventa) solicita un producto, los operarios lo buscan en los estantes, lo recogen y lo envían a su destino.
En un muelle de intercambio rápido (*crossdock*), los bienes que están llegando desde un proveedor tienen asignado ya un cliente, de manera que los operarios únicamente mueven el embarque del vehículo que llega hasta el vehículo que sale con el destino apropiado para el cliente. No es difícil de imaginar que uno de los mayores requisitos para que el muelle de intercambio rápido funcione bien, sea un sistema de información efectivo que soporte la operación.

*¿Cuál es la diferencia entre un muelle de intercambio rápido (*crossdock*) con predistribución y uno con posdistribución?*

Una forma de operación para clasificar en un muelle de intercambio rápido (*crossdock*) es asignar el cliente a una estiba específica o una determinada zona o sitio para colocar allí, temporalmente, el producto o paquete que viene destinado a él. En una operación de *crossdocking con predistribución* el *cliente es asignado antes* de que el embarque salga del proveedor, de manera que tan pronto llegue el embarque al *crossdock*, éste se coloca en el sitio previamente asignado y se etiqueta para su transferencia.
En una operación de *crossdocking con posdistribución,* los operarios del *crossdock asignan aleatoriamente* los materiales a las zonas de almacenamiento temporal existentes.
La *predistribución* es definitivamente más difícil de implementar debido a que los vendedores del *crossdock* deben conocer los clientes a los que habrá que separarles espacio en el muelle, antes de que se envíen los pedidos. Esto requiere una gran cantidad de información que se transfiere para lograr una coordinación adecuada de las operaciones. Para un distribuidor con varios cientos de vendedores, este problema es muy, pero muy grande y necesita como soporte un sistema de información centralizado con mucha capacidad y agilidad en el tiempo de respuesta.

¿Cuáles son algunos requisitos principales para implementar un sistema de crossdocking?

El primer requisito es la existencia de una red logística que una a los clientes que envían mercancías con el centro de *crossdocking* y a éste con los destinatarios de ellas. Esta red puede estar estructurada de la siguiente manera: nodos de producción o recolección en cada ciudad, transportes entre las ciudades, nodos de clasificación o inventarios en cada ciudad, transporte de distribución y recolección al interior de las ciudades. La red de flujo debe ser básicamente constante, o sea que la demanda debe ser estable y suficiente para mantenerla funcionando permanentemente.
El segundo requisito es un sistema centralizado de información, que controle el estado de los pedidos y de los transportes.
El tercer requisito es un sistema de embalaje modular, que facilite agregar o quitar paquetes a las estibas, cuando los pedidos grandes se partan en otros más pequeños, o a la inversa.

Manuel escuchó atentamente el informe del ingeniero y se felicitó mentalmente por haber escogido un profesional con el perfil de Mario, experto en "sistemas de información", porque según lo que había entendido, uno de los requisitos para implementar *la técnica del crossdocking* era tener en marcha un excelente sistema de información.

La segunda tarea que Manuel asignó a Mario fue demostrar a los demás accionistas que la implementación de un muelle de intercambio rápido podría bajar los costos de almacenamiento en proporciones considerables, y mejorar el tiempo de respuesta a los clientes.

Mario prácticamente no durmió esa noche pensando en una posible solución a la tarea que le habían asignado. El no podía descuidar la operación diaria de la bodega ni podía ensayar la operación en forma de muelle de intercambio rápido sin causar retrasos en las entregas. Al otro día comentó con Sonia su preocupación. Ella sonrió y le dijo: " Y para qué aprendió simulación?" Inmediatamente a Mario se le "iluminó" la cara, porque a él había disfrutado mucho su curso de simulación y había aprendido a trabajar con el software animado de Promodel®. "Gracias Sonia", fue lo único que se le ocurrió decir.

Para Mario era evidente que al aplicar *crossdocking* los productos no se almacenaban, sino que se transbordaban para ser remitidos al cliente. Con este sistema, los costos de mantenimiento de inventario se reducían a cero (los de ordenar seguirían siendo los mismos en las dos alternativas). Los

tiempos de transporte entre los proveedores y el centro de distribución seguían siendo los mismos, e igual sucedía con los tiempos de transporte entre el centro de distribución y el cliente, que permanecían iguales, de tal manera que la diferencia estaba en los tiempos de almacenamiento. Bastaría entonces calcular los tiempos para la alternativa del *crossdocking*, porque los de la otra alternativa (almacenar) siempre serían mayores.

A continuación, se presenta el modelo elaborado por el ingeniero Fajardo para calcular los tiempos de *crossdocking*.

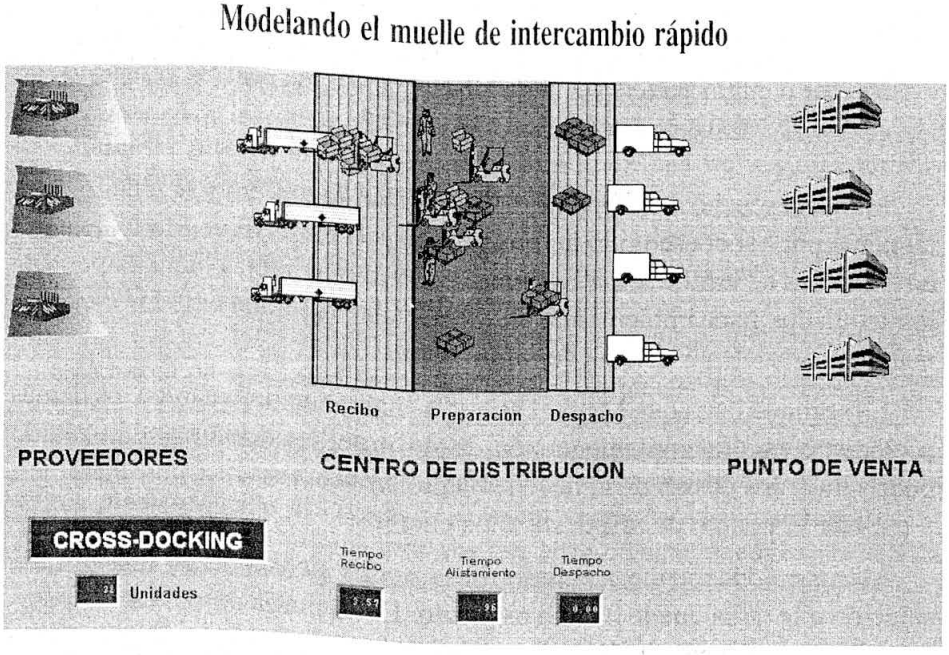


Figura 4. 24 *Layout* del muelle de intercambio rápido.

1. *BUILD (Locations)*

Se incluyeron tres *fábricas* como proveedoras y cuatro *puntos de venta* como clientes. El centro de distribución se modeló con tres *muelles de recibo*, tres zonas de *recibo*, cuatro *muelles de despacho*, cuatro zonas de *consolidación* y doce zonas de *preparación*. Ver figura 4.33 para un mayor detalle de las zonas.

Name	Cap.	Units	DTS	Statistics	Rules
Fábrica_uno	Inf.	1	None	Time series	Oldest
Fábrica_dos	Inf.	1	None	Time series	Oldest
Fábrica_tres	Inf.	1	None	Time series	Oldest
Punto_Venta_uno	Inf.	1	None	Time series	Oldest
Punto_Venta_dos	Inf.	1	None	Time series	Oldest
Punto_Venta_tres	Inf.	1	None	Time series	Oldest
Punto_Venta_cuatro	Inf.	1	None	Time series	Oldest
Muelle_Recibo_uno	1	1	None	Time series	Oldest
Muelle_Recibo_dos	1	1	None	Time series	Oldest
Muelle_Recibo_tres	1	1	None	Time series	Oldest
Recibo_uno	Inf.	1	None	Time series	Oldest
Recibo_dos	Inf.	1	None	Time series	Oldest
Recibo_tres	Inf.	1	None	Time series	Oldest
Muelle_Despacho_uno	1	1	None	Time series	Oldest
Muelle_Despacho_dos	1	1	None	Time series	Oldest
Muelle_Despacho_tres	1	1	None	Time series	Oldest
Muelle_Despacho_cuatro	1	1	None	Time series	Oldest
Preparación_Uno_a	Inf.	1	None	Time series	Oldest
Preparación_Dos_a	Inf.	1	None	Time series	Oldest
Preparación_Tres_a	Inf.	1	None	Time series	Oldest
Preparación_Cuatro_a	Inf.	1	None	Time series	Oldest
Preparación_Uno_b	Inf.	1	None	Time series	Oldest
Preparación_Dos_b	Inf.	1	None	Time series	Oldest
Preparación_Tres_b	Inf.	1	None	Time series	Oldest
Preparación_Cuatro_b	Inf.	1	None	Time series	Oldest
Preparación_Uno_c	Inf.	1	None	Time series	Oldest
Preparación_Dos_c	Inf.	1	None	Time series	Oldest
Preparación_Tres_c	Inf.	1	None	Time series	Oldest
Preparación_Cuatro_c	Inf.	1	None	Time series	Oldest
Consolidación_uno	Inf.	1	None	Time series	Oldest
Consolidación_dos	Inf.	1	None	Time series	Oldest
Consolidación_tres	Inf.	1	None	Time series	Oldest
Consolidación_cuatro	Inf.	1	None	Time series	Oldest

2. *BUILD (Entities)*

Entities [6]

Icon	Name	Speed (mpm)	Stats...	Notes...
	Proveedor_Uno	50	Time Series	
	Proveedor_dos	50	Time Series	
	Proveedor_tres	50	Time Series	
	Caja_uno	50	Time Series	
	Producto_Paletizado	50	Time Series	
	carga_uno	50	Time Series	

Figura 4. 25 Entidades que fluyen desde los proveedores hasta los puntos de venta.

3. *BUILD (Arrivals)*

Arrivals [4]

Entity...	Location...	Qty each...	First Time	Occurrences	Frequency	Logic	Disable
Proveedor_Uno	Fabrica_uno	1	Fri, Mar 29 200	inf	E(20)		No
Proveedor_Uno	Fabrica_dos	1	Fri, Mar 29 200	inf	n(18,3)		No
Proveedor_Uno	Fabrica_tres	1	Fri, Mar 29 200	inf	E(15)		No

Figura 4.26 Llegadas al sistema.

4. *BUILD (Attributes)*

Attributes [3]

ID	Type...	Classification...	Notes...
provi	Real	Ent	
prov2	Real	Ent	
prov3		Ent	

Figura 4.27 Atributos del modelo.

5. *BUILD (Variables)*

Variables (global) [5]

Icon	ID	Type...	Initial value	Stats...	Notes...
Yes	cantidad_prod	Integer	0	Time Serie:	
Yes	Tiempo_de_Recibo	Real	0	Time Serie:	
Yes	Tiemo_de_alistamiento	Real	0	Time Serie:	
Yes	tiempo_de_despacho	Real	0	Time Serie:	

Figura 4.28 Variables del modelo.

6. *BUILD (Path Networks)*

Path Networks [8]

Graphic	Name	Type	T/S	Paths...	Interfaces...	Mapping...	Nodes
	fab_uno	Passing	Speed & Dista	1	2	0	2
	fab_dos	Passing	Speed & Dista	1	2	0	2
	fab_tres	Passing	Speed & Dista	1	2	0	2
	distribucion	Passing	Speed & Dista	23	19	0	19
	distri_uno	Passing	Speed & Dista	1	2	0	2
	distri_dos	Passing	Speed & Dista	1	2	0	2
	distri_tres	Passing	Speed & Dista	1	2	0	2
	distri_cuatro		Speed & Dista	1	2	0	2

Figura 4.29 Redes del modelo.

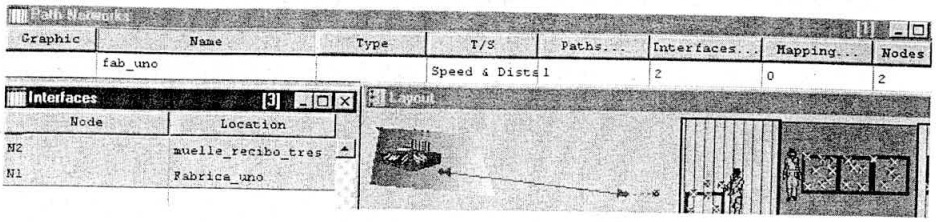


Figura 4.30 Red *fab-uno* con sus interfaces.

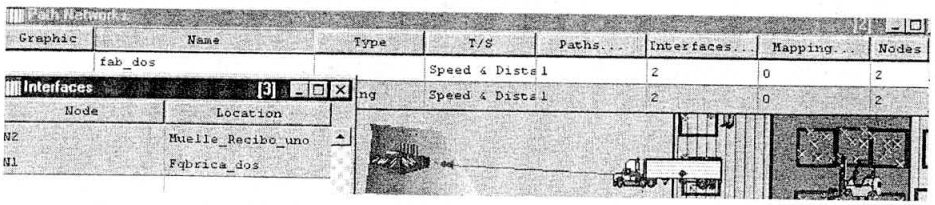


Figura 4.31 Red *fab-dos* con sus interfaces.

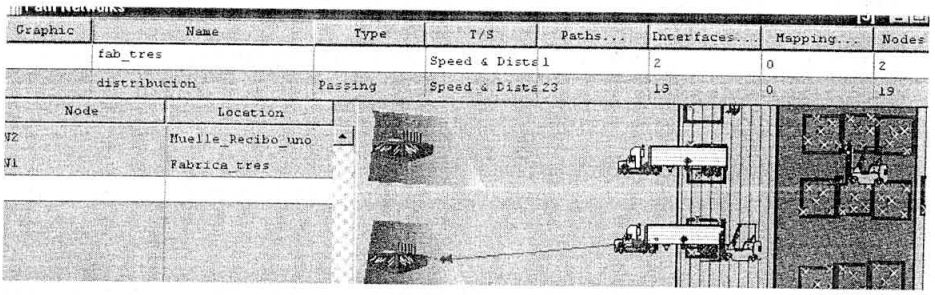


Figura 4.32 Red *fab-tres* con sus interfaces.

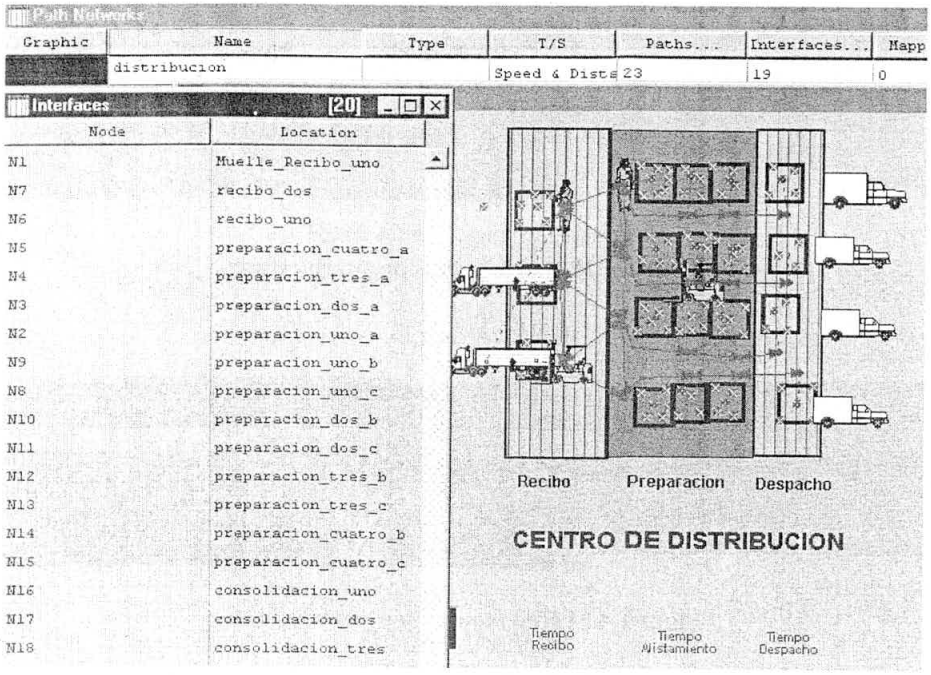


Figura 4.33 Red *Distribución* y sus interfaces.

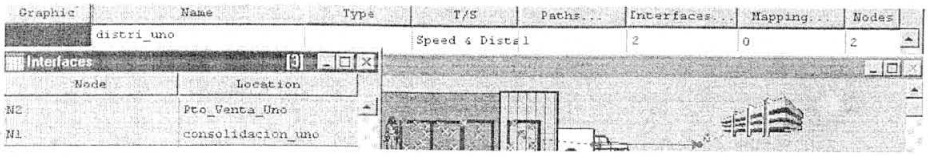


Figura 4.34 Red *Distri_uno* y sus interfaces.

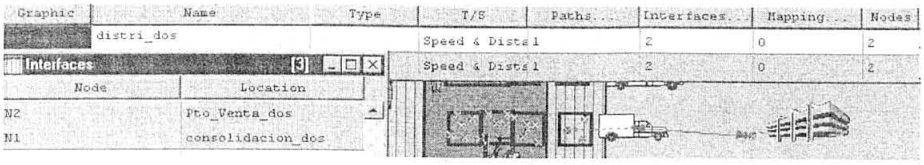


Figura 4.35 Red *Distri_dos* y sus interfaces.

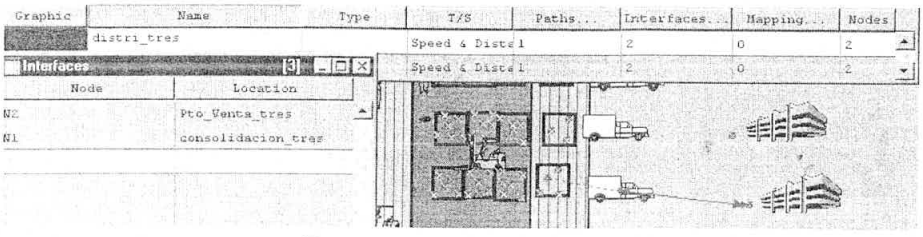


Figura 4.36 Red *Distri_tres*.

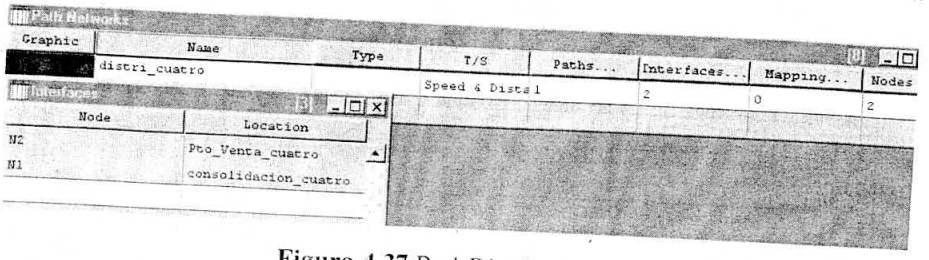


Figura 4.37 Red *Distri_cuatro.*

7. *BUILD (Resources)*

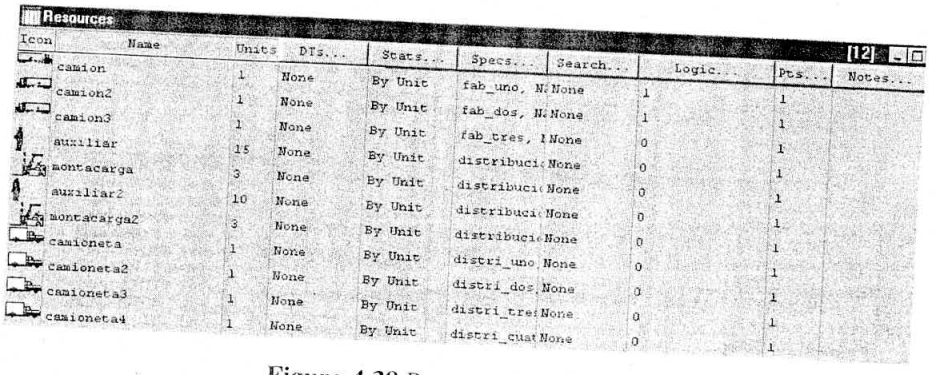


Figura 4.38 Recursos del modelo.

Resources *(Specifications)*

Name	Path Network	Home	Return if Idle	Entity Search
camión	fab_uno	N2	SI	Longest waiting
camión 2	fab_dos	N2	SI	Longest waiting
camión 3	fab_tres	N2	SI	Longest waiting
auxiliar	distribución	N1	NO	Longest waiting
montacarga	distribución	N6	NO	Longest waiting
auxiliar 2	distribución	N2	NO	Longest waiting
montacarga 2	distribución	N10	NO	Longest waiting
camioneta	distri_uno	N1	SI	Longest waiting
camioneta 2	distri_dos	N1	SI	Longest waiting
camioneta 3	distri_tres	N1	SI	Longest waiting
camioneta 4	distri_cuatro	N1	SI	Longest waiting

8. *BUILD (Processing)*

Entity	Location	Operation	Output	Destination	Rule	Move logic
Proveedor_uno	Fábrica_uno		Proveedor_uno	Muelle_recibo_tres	First 1	MOVE WITH camion THEN FREE
Proveedor_uno	Muelle_recibo_tres	prov1 = CLOCK(MIN)	Caja_uno Proveedor_uno	Recibo_tres EXIT	First 50 First 1	
ALL	Recibo_tres	WAIT p(3,1) INC cantidad_prod	ALL ALL ALL ALL	Preparación_uno_a Preparación_dos_a Preparación_tres_a Preparación_cuatro_a	0.25 0.25 0.25 0.25	MOVE WITH auxiliar, 1 or auxiliar2 THEN FREE Tiempo_de_Recibo = CLOCK(MIN)-prov1 MOVE WITH auxiliar, 1 THEN FREE Tiempo_de_Recibo = CLOCK(MIN)-prov1 MOVE WITH auxiliar, 1 THEN FREE Tiempo_de_Recibo = CLOCK(MIN)-prov1 MOVE WITH montacarga THEN FREE Tiempo_de_Recibo = CLOCK(MIN)-prov1
Proveedor_uno	Fábrica_dos	INC cantidad_prod	Proveedor_uno	Muelle_recibo-dos	First 1	MOVE WITH camion2 THEN FREE
Proveedor_uno	Muelle_recibo-dos		Proveedor_dos Proveedor_uno	Recibo_dos EXIT	First 50 First 1	
ALL	Recibo_dos	INC cantidad_prod	ALL ALL ALL ALL	Preparación_uno_b Preparación_dos_b Preparación_tres_b Preparación_cuatro_b	0.4 0.1 0.25 0.25	MOVE WITH montacarga, 1 or auxiliar2, 2 THEN FREE MOVE WITH auxiliar or auxiliar2, 2 THEN FREE MOVE WITH auxiliar or auxiliar2, 2 THEN FREE MOVE WITH auxiliar or auxiliar2, 2 THEN FREE
Proveedor_uno	Fábrica_tres		Proveedor_uno	Muelle_recibo-uno	First 1	MOVE WITH camion3 THEN FREE
Proveedor_uno	Muelle_recibo-uno		Proveedor_uno Proveedor_tres	EXIT Recibo_uno	First 1 First 50	
ALL	Recibo_uno	INC cantidad_prod	ALL ALL ALL ALL	Preparación_uno_c Preparación_dos_c Preparación_tres_c Preparación_cuatro_c	0.1 0.2 0.2 0.5	MOVE WITH auxiliar THEN FRE MOVE WITH auxiliar THEN FREE MOVE WITH auxiliar THEN FREE MOVE WITH auxiliar THEN FREE
ALL	Preparación_uno_a	ACCUM 2 GROUP 2 AS Producto_Paletizado				
ALL	Preparación_dos_a	ACCUM 4 GROUP 4 AS Producto_Paletizado				
ALL	Preparación_tres_a	ACCUM 3 GROUP 3 AS Producto_Paletizado				
ALL	Preparación_cuatro_a	ACCUM 5 GROUP 5 AS Producto_Paletizado				
ALL	Preparación_uno_b	ACCUM 2 GROUP 2 AS Producto_Paletizado				
ALL	Preparación_dos_b	ACCUM 3 GROUP 3 AS Producto_Paletizado				
ALL	Preparación_tres_b	ACCUM 10 GROUP 10 AS Producto_Paletizado				
ALL	Preparación_cuatro_b	ACCUM 15 GROUP 15 AS Producto_Paletizado				
ALL	Preparación_uno_c	ACCUM 5 GROUP 5 AS Producto_Paletizado				
ALL	Preparación_dos_c	ACCUM 5 GROUP 5 AS Producto_Paletizado				

ALL	Preparación_tres_c	ACCUM 4 GROUP 4 AS Producto_Paletizado				
ALL	Preparación_cuatro_c	ACCUM 4 GROUP 4 AS Producto_Paletizado				
Producto_paletizado	Preparación_uno_a	prov2 = CLOCK(MIN)	Producto_paletizado	Consolidación_uno	First 1	MOVE WITH montacarga2 THEN FREE DEC cantidad_prod Tiempo_de_alistamiento=clock(min)-prov2 prov1 = CLOCK(MIN)
Producto_paletizado	Preparación_uno_b		Producto_paletizado	Consolidación_uno	First 1	MOVE WITH montacarga2 THEN FREE DEC cantidad_prod
Producto_paletizado	Preparación_uno_c		Producto_paletizado	Consolidación_uno	First 1	MOVE WITH montacarga2 THEN FREE DEC cantidad_prod
Producto_paletizado	Preparación_dos_a		Producto_paletizado	Consolidación_dos	First 1	MOVE WITH montacarga2 THEN FREE DEC cantidad_prod
Producto_paletizado	Preparación_dos_b		Producto_paletizado	Consolidación_dos	First 1	MOVE WITH montacarga2 THEN FREE DEC cantidad_prod
Producto_paletizado	Preparación_dos_c		Producto_paletizado	Consolidación_dos	First 1	MOVE WITH montacarga2 THEN FREE DEC cantidad_prod
Producto_paletizado	Preparación_tres_a		Producto_paletizado	Consolidación_tres	First 1	MOVE WITH montacarga2 THEN FREE DEC cantidad_prod
Producto_paletizado	Preparación_tres_b		Producto_paletizado	Consolidación_tres	First 1	MOVE WITH montacarga2 THEN FREE DEC cantidad_prod
Producto_paletizado	Preparación_tres_c		Producto_paletizado	Consolidación_tres	First 1	MOVE WITH montacarga2 THEN FREE DEC cantidad_prod
Producto_paletizado	Preparación_cuatro_a		Producto_paletizado	Consolidación_cuatro	First 1	MOVE WITH montacarga2 THEN FREE DEC cantidad_prod
Producto_paletizado	Preparación_cuatro_b		Producto_paletizado	Consolidación_cuatro	First 1	MOVE WITH montacarga2 THEN FREE DEC cantidad_prod
Producto_paletizado	Preparación_cuatro_c		Producto_paletizado	Consolidación_cuatro	First 1	MOVE WITH montacarga2 THEN FREE DEC cantidad_prod
ALL	Consolidación_uno	ACCUM 5 GROUP 5 AS Carga_uno				
Carga_uno	Consolidación_uno	prov3 = CLOCK(MIN)	Carga_uno	Pto_Venta_uno	First 1	MOVE WITH camioneta THEN FREE tiempo_de_despacho=clock(min)-prov3
ALL	Consolidación_dos	ACCUM 5 GROUP 5 AS Carga_uno				
Carga_uno	Consolidación_dos		Carga_uno	Pto_Venta_dos	First 1	MOVE WITH camioneta2 THEN FREE
ALL	Consolidación_cuatro	ACCUM 5 GROUP 5 AS Carga_uno				
Carga_uno	Consolidación_cuatro		Carga_uno	Pto_Venta_cuatro	First 1	MOVE WITH camioneta4 THEN FREE
ALL	Consolidación_tres	ACCUM 6 GROUP 6 AS Carga_uno				
Carga_uno	Consolidación_tres		Carga_uno	Pto_Venta_tres	First 1	MOVE WITH camioneta3 THEN FREE
ALL	Pto_Venta_uno		ALL	EXIT	First 1	
ALL	Pto_Venta_dos		ALL	EXIT	First 1	
ALL	Pto_Venta_tres		ALL	EXIT	First 1	
ALL	Pto_Venta_cuatro		ALL	EXIT	First 1	
Proveedor_un	Preparación_d		Proveedor	Preparación_dos_a	First 1	

9. *SIMULATION (Options)*

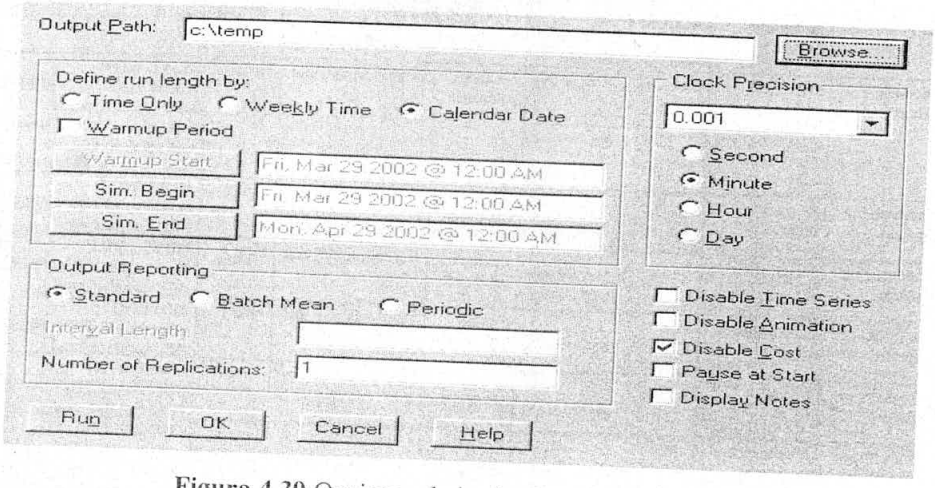
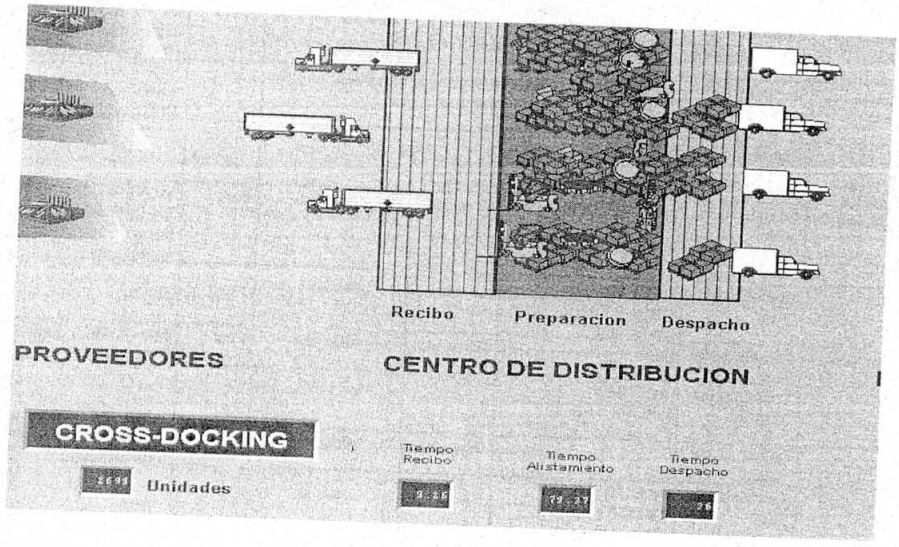


Figura 4.39 Opciones de la simulación del modelo.

La simulación se corrió con 1000 llegadas de cada una de las tres fábricas.

10. ANÁLISIS DE LOS RESULTADOS



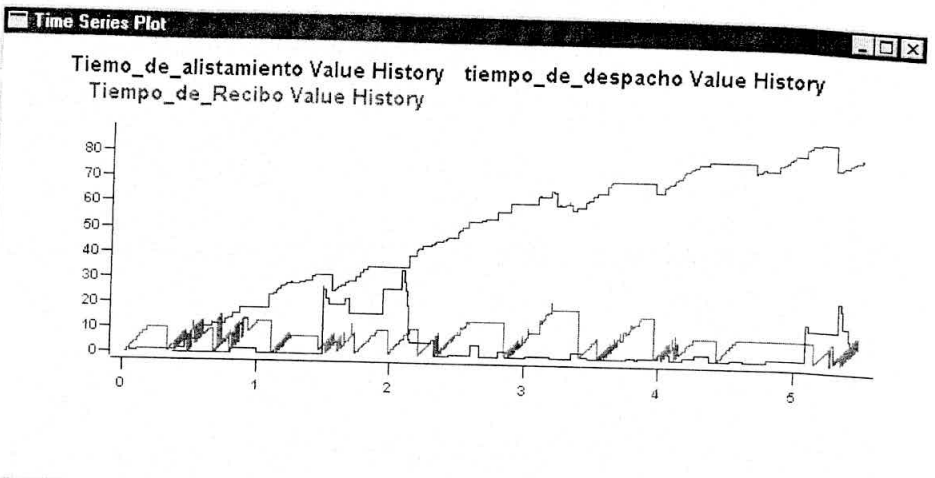


Figura 4.40 Comportamiento de los tiempos de alistamiento, despacho y recibo.

Caso 5. Centro de distribución de mercancías

Como se comentó en el Caso 1, ServiECI es una empresa de servicios dedicada a la recolección y entrega de mercancías, en las principales ciudades de Colombia. En Bogotá, la empresa tiene una bodega central, o centro de distribución, que es un lugar de paso de la mercancía a donde llegan, por una parte, grandes tractomulas y camiones, llamados genéricamente *vanes*, que normalmente viajan de noche, y cubren la geografía nacional; y por otra, *furgones* que son camiones pequeños que normalmente viajan de día, dentro de la ciudad. La tripulación de los vehículos varía entre dos y tres personas.

Tan pronto llega un *van* a la bodega, se le asigna un equipo de operarios para descargarlo y volverlo a cargar; así como una zona demarcada en el piso, en la llamada zona de embarque subdividida en bloques según las ciudades de destino. A la ubicación de mercancías en su correspondiente zona de destino se le llama *Zonificación*.

Durante el día los operarios cargan los *furgones* una vez salen los *vanes* a sus destinos. Los *furgones*, una vez cargados, salen a entregar las mercancías a los domicilios previamente zonificados; cuando terminan la entrega, pasan por las oficinas locales de ServiECI en Bogotá, recogen las mercancías y se dirigen al centro de distribución, donde son descargados nuevamente y salen para mantenimiento. Regresan al otro día. Los operarios zonifican la mercancía que han descargado de los *furgones*, para cargarla luego en los *vanes*.

Guerrero, el gerente de operaciones, escogió a Darío Saavedra, ingeniero industrial, encargado de mejorar la productividad de los procesos, para analizar cómo se realiza la operación del centro de distribución de Bogotá, identificar los cuellos de botella, hacer un diagnóstico de la operación y proponer alternativas de mejoramiento.

Darío empleó varios días en conocer la operación del centro de distribución e imaginarla como un conjunto de procesos; identificar la información que fluye, sus usuarios; recolectar los datos del diseño de instalaciones, número de operarios, porcentajes de productos que pueden colocarse en estibas, número de unidades recogidas y distribuidas, hora de llegada y salida de los diferentes tipos de vehículos, número de unidades en inventario, etcétera.

El ingeniero Saavedra delimitó dos grandes procesos: recolección y distribución.

El proceso de recolección lo hacen las oficinas locales, a donde llegan los usuarios a enviar sus paquetes, y los *furgones* a trasladar las mercancías hasta el centro de distribución. En este proceso de recolección se controlan las mercancías, los dineros, los *furgones* y las tripulaciones. En cada oficina local, se elabora un documento, conocido como *la guía*. Allí se relacionan las mercancías, sus destinos, el costo del envío, las características especiales de transporte, la hora de llegada del vehículo y los nombres de los componentes de la tripulación. Esta información se introduce en el sistema de información logístico de la empresa.

El proceso de distribución se inicia en el centro de distribución y termina en los usuarios. En este proceso se controlan las mercancías, los dineros, los vehículos (*vanes y furgones*), las tripulaciones, los operarios de cargue y descargue, los inventarios temporales, las quejas y el servicio al cliente. La información de este proceso también se introduce en el sistema de información logístico de la empresa.

En el poco tiempo de este rápido y forzado aprendizaje, Darío había comenzado a pensar en su modelo de simulación, a definir las instalaciones, las entidades y sus llegadas y, con la nueva información, había comenzado a definir el proceso. En la operación del sistema, descubrió dos grandes procesos: recolección y distribución.

Para el modelo analizó la operación de los centros de distribución que operan en las ciudades grandes, y escogió el de Bogotá como prototipo para simular su operación interna. Tanto en el proceso de recolección como en el de distribución, los dos tipos de vehículos mencionados unen las fábricas con los sitios de distribución final. Para modelar la operación interna del centro de distribución de Bogotá, hizo las mediciones necesarias para que la distribución (*layout*) quedara a escala. Pidió información sobre los turnos, el número de operario y el horario de trabajo, para incluir esta información en el modelo. Lo mismo hizo con los vehículos y las tripula-

ciones, horarios, documentos, información, entre otros. También averiguó cómo funcionaba el sistema de información, tanto en la entrada y salida de datos como en el control y verificación de su validez.

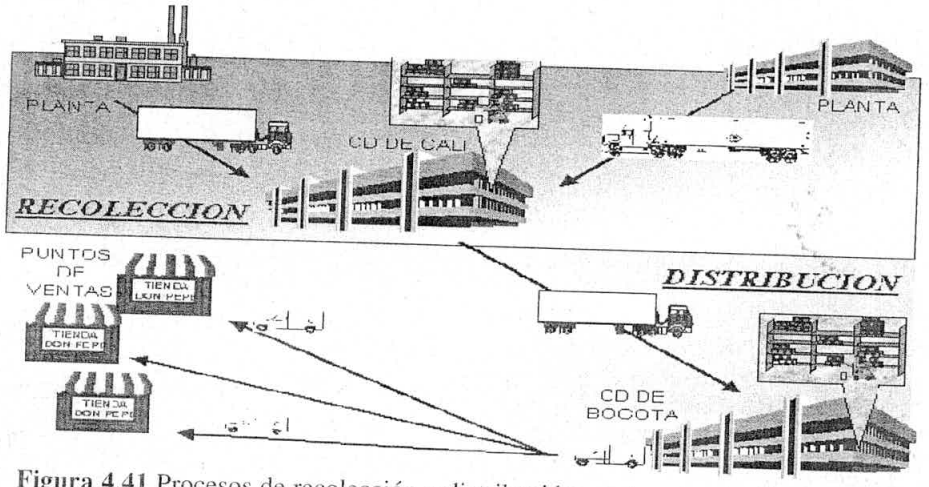


Figura 4.41 Procesos de recolección y distribución en el centro de distribución.

A través de entrevistas con las personas encargadas de la programación y el control de la operación, se informó de los detalles y de cómo se estaban programando los recorridos de los vehículos en las diferentes rutas, nacionales o locales, y tomó nota de los indicadores que se estaban utilizando para medir la productividad de los operarios y de los vehículos.

Cuando terminó el modelo, pensó que era hora de verificar si su representación ficticia se parecía a la situación real o no, de manera que simuló la operación de un día con volumen de mercancías representativo. Cuando estuvo seguro de que su modelo funcionaba, pidió al gerente Guerrero que citara a una reunión con ingenieros y supervisores para mostrarles el modelo y que ellos pudiesen hacer las observaciones correspondientes. Así se hizo. A Darío le fue muy bien en la presentación, y lo mejor de todo fue la participación del grupo.

Con base en las observaciones que le hicieron, Darío hizo los ajustes y preparó la simulación de una semana de alto movimiento. Esta simulación la mostró a los ingenieros y al gerente Guerrero. Este último cambió completamente su actitud, hasta ahora de escepticismo, por otra analítica y de cooperación. De aquí en adelante, los cambios al modelo fueron sugeridos por el equipo directivo de operaciones de ServiECI. El modelo se fue me-

jorando poco a poco con la colaboración del equipo directivo. Se hicieron nuevos ensayos y, a medida que se generó credibilidad en la simulación, se buscaron nuevos retos.

Las tres partes principales del modelo

Darío, como se comentó, pensó en dos grandes procesos: recolección y distribución. El modelo que construyó constaba de tres grandes módulos o partes: a) operación del muelle en el centro de distribución de Bogotá; b) centro de llamadas del sistema de información y, c) rutas de distribución de la zona norte de Bogotá.

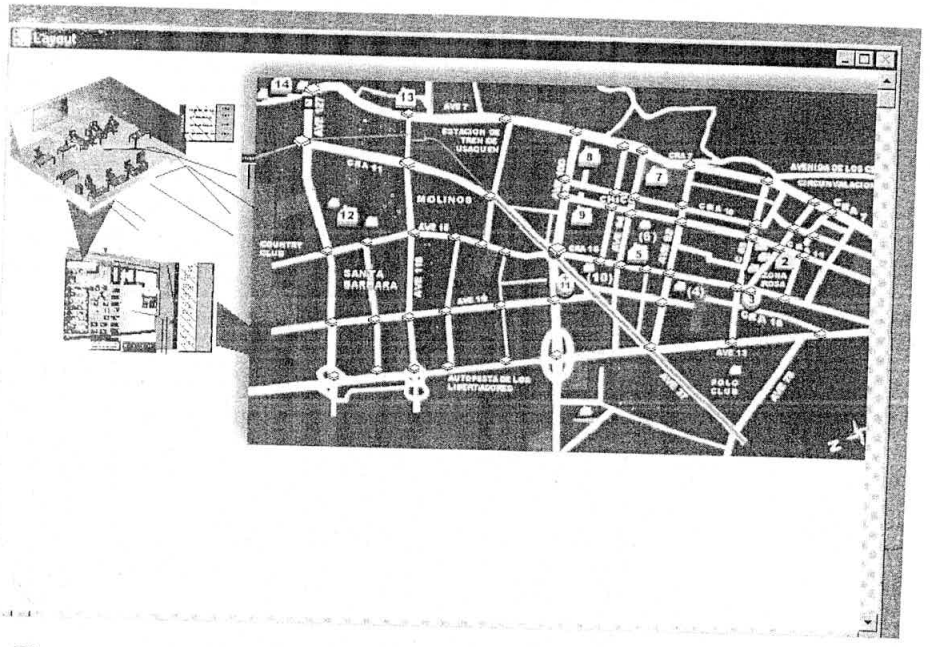


Figura 4.42 Vista integral del *layout* del modelo. En la parte superior izquierda está el centro de llamadas; abajo, el muelle del centro de distribución y, a la izquierda, el mapa de rutas en la zona norte de Bogotá.

Es muy complicado presentar completo el modelo construido porque tiene ciento cuarenta instalaciones, treinta entidades, cuatro redes de movimiento de cincuenta nodos cada una, ochenta y ocho variables. Se presentan a continuación gráficas de sus módulos principales:

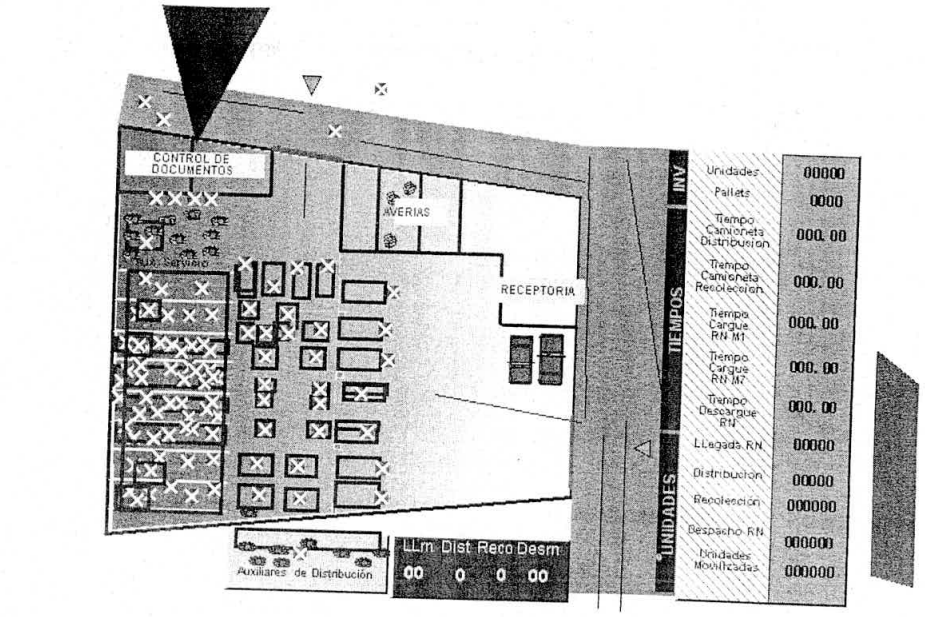


Figura 4.43 *Layout* del muelle del centro de distribución de Bogotá. Abajo la zona en donde los operarios esperan la llegada de los vehículos y arriba, a la izquierda, el control de documentación.

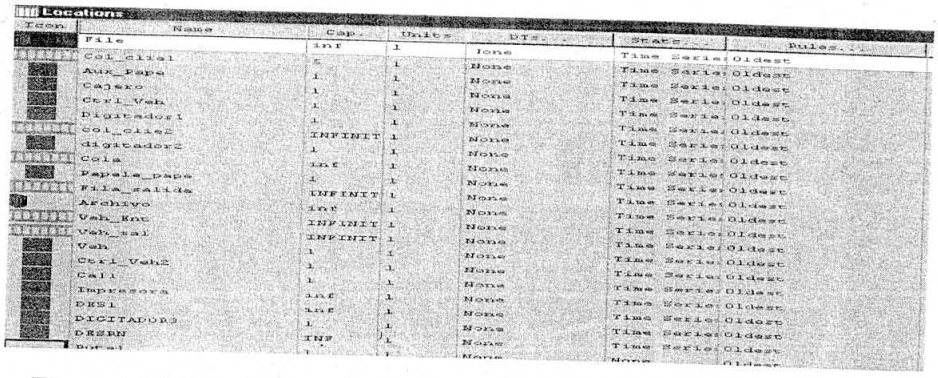
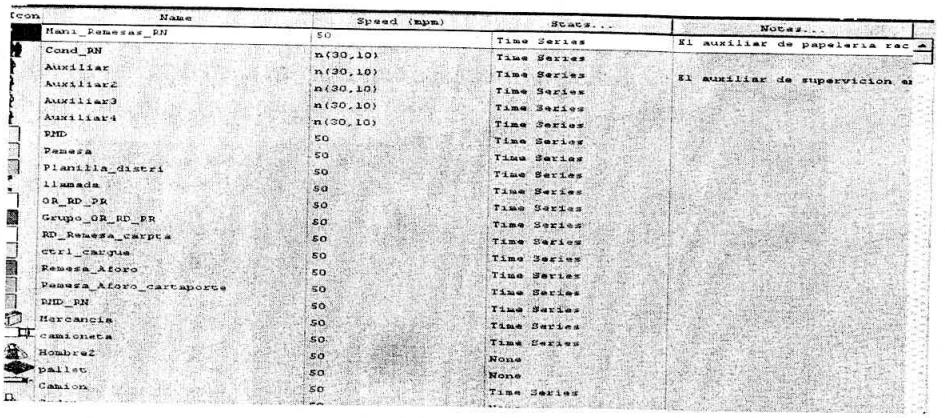


Figura 4.44 Algunas de las instalaciones del modelo correspondientes al muelle del centro de distribución.



Icon	Name	Speed (mpm)	Stats...	Notes...
	Mant_Remesas_RN	50	Time Series	El auxiliar de papeleria rec
	Cond_RN	n(30,10)	Time Series	
	Auxiliar	n(30,10)	Time Series	El auxiliar de supervicion es
	Auxiliar2	n(30,10)	Time Series	
	Auxiliar3	n(30,10)	Time Series	
	Auxiliar4	n(30,10)	Time Series	
	RMD	50	Time Series	
	Remesa	50	Time Series	
	Planilla_distri	50	Time Series	
	llamada	50	Time Series	
	OR_RD_PR	50	Time Series	
	Grupo_OR_RD_PR	50	Time Series	
	RD_Remesa_carpta	50	Time Series	
	ctrl_cargue	50	Time Series	
	Remesa_Aforo	50	Time Series	
	Remesa_Aforo_cartaporte	50	Time Series	
	RMD_RN	50	Time Series	
	Mercancia	50	Time Series	
	camioneta	50	Time Series	
	Hombre2	50	None	
	pallet	50	None	
	Camion	50	Time Series	

Figura 4.45 Algunas de las entidades consideradas en el modelo.

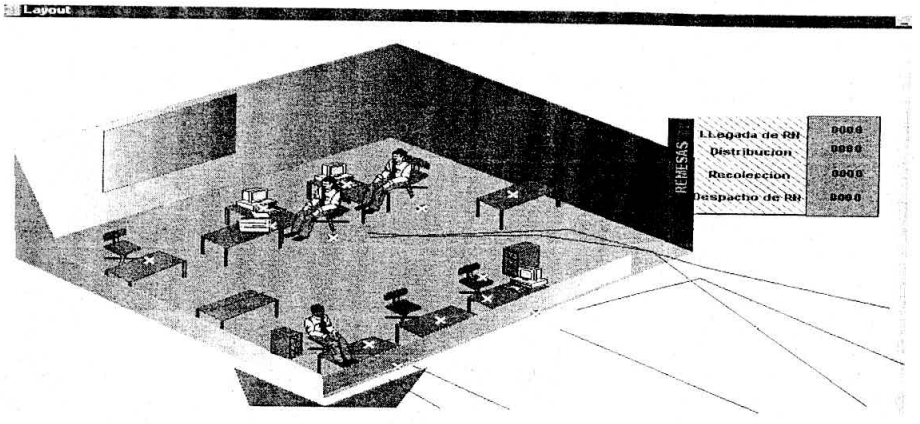


Figura 4.46 Centro de control y de llamadas del sistema de información.

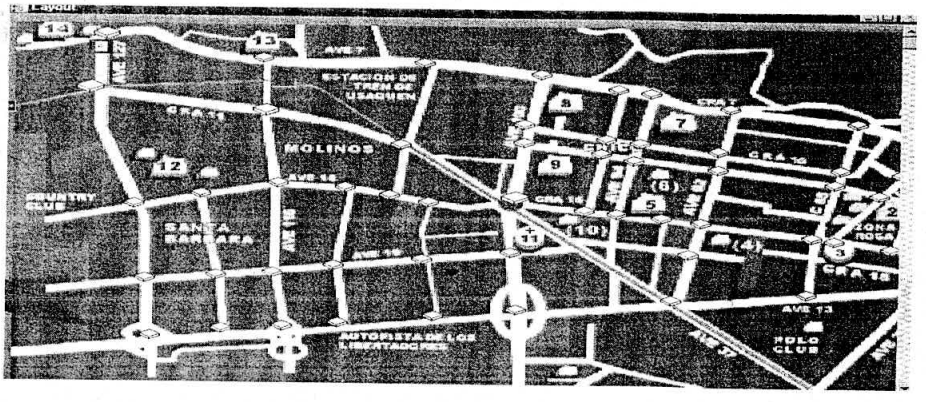


Figura 4.47 Rutas de recolección y distribución en la zona norte de Bogotá.

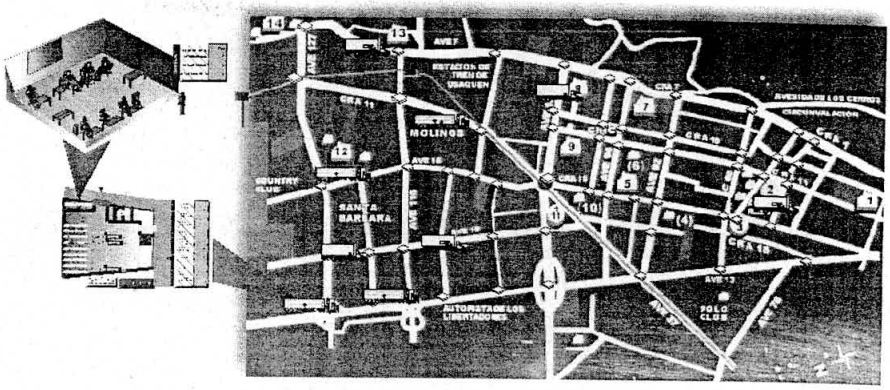


Figura 4.48 Vista integral del modelo en el momento de la simulación.

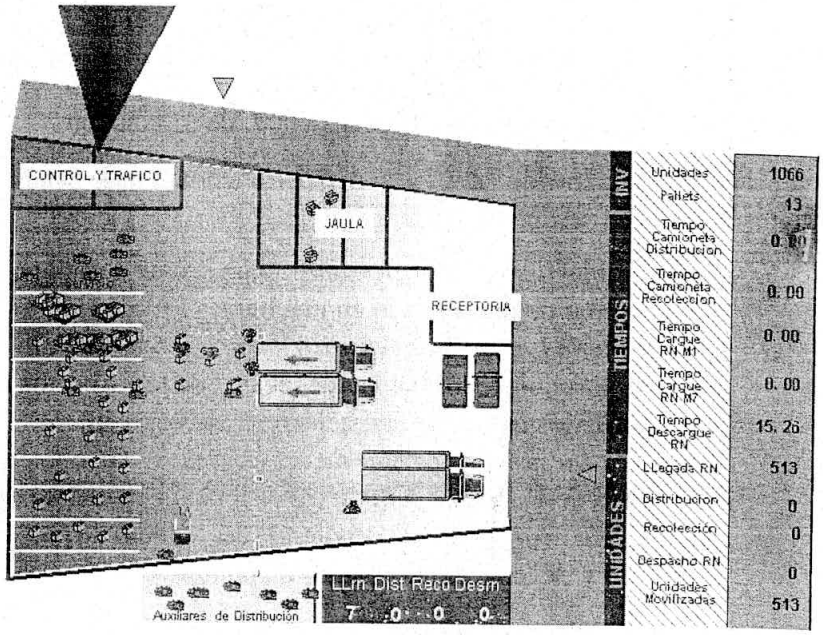


Figura 4.49 Operación del muelle del centro de distribución durante la simulación.

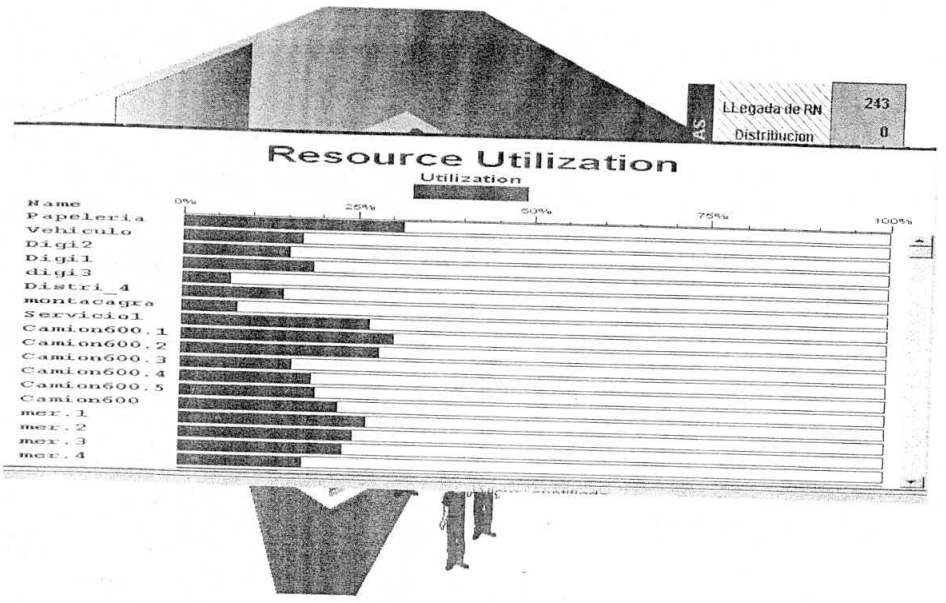


Figura 4.50 Operación del sistema de información y control durante la simulación.

Diagnóstico de la operación del centro de distribución

Para analizar la operación del Centro de Distribución, Darío estableció los siguientes indicadores de desempeño:

1. *Unidades movilizadas por personas*: Número de cajas que moviliza un auxiliar de distribución en una hora.
2. *Utilización de los vehículos de distribución y recolección*: Porcentaje de utilización de los vehículos (es uno de los aspectos más importantes de los costos de operación en este negocio).
3. *Tiempo de cargue y descargue de los vehículos:* Tiempo empleado en cargar o descargar un camión (es otro factor crítico de esta operación). Este tiempo varía dependiendo del número de personas que se utilicen y de los recursos físicos disponibles.

Unidades movilizadas por persona

Darío encontró que el número promedio de unidades movilizadas por persona estaba por debajo del estándar, lo que significaba que los auxiliares de distribución de Bogotá estaban trabajando por debajo de su capacidad. De acuerdo con sus observaciones, las posibles causas de la baja productivi-

dad del personal podrían ser la falta de motivación del personal y una incorrecta programación de los turnos, ocasionada por la mala sincronización entre la hora de entrada de los operarios y la hora de llegada de los vehículos de ruta nacional, agravada ésta por el retraso de los vehículos que no llegaban a la hora esperada al centro de distribución.

El mayor volumen de movimiento de mercancía ocurría entre las diez de la mañana y las dos de la tarde, a pesar de que los *vanes* debían estar descargando teóricamente entre las seis y las diez de la mañana. Con base en la hipótesis teórica se programaba un turno de auxiliares de distribución que iniciaba su jornada a las 6:00 a.m. y que, al no encontrar vehículos para descargar, descansaba casi cuatro horas, ocasionando una baja considerable en el indicador de productividad.

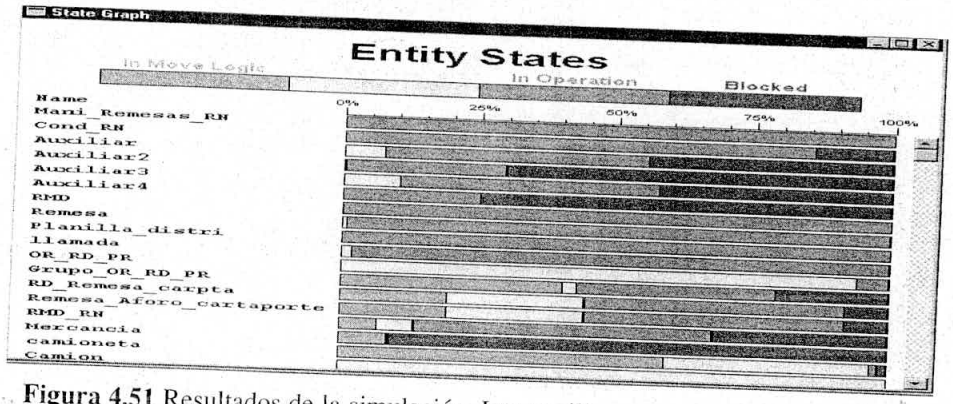


Figura 4.51 Resultados de la simulación. Los auxiliares permanecían en operación menos del cincuenta por ciento.

Tiempo de cargue y descargue de los vehículos

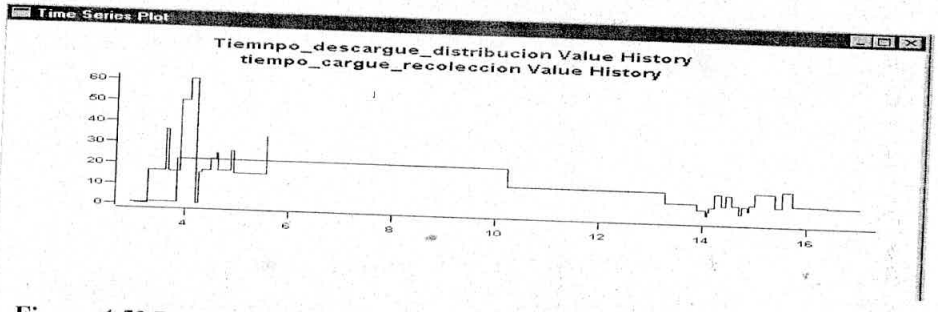


Figura 4.52 Resultados de la simulación: tiempo de descargue para el proceso de distribución y tiempo de cargue para recolección.

Los tiempos promedios de cargue y descargue que Darío encontró en la simulación, comparados con los tiempos meta esperados, fueron:

Vehículo	Tiempo actual	Tiempo meta
Furgón	53 minutos	20 minutos
Van	105 minutos	50 minutos

De acuerdo con lo observado por Darío, el cargue y descargue de un furgón era demasiado lento con respecto al tiempo óptimo, debido a que en ocasiones los muelles se encontraban copados en el momento de la llegada de los vehículos al centro y tenían que esperar en cola durante treinta minutos, aproximadamente, antes de empezar la operación.

En cuanto a los *vanes*, también era lento el tiempo de descargue con respecto a lo esperado. Esto era ocasionado por la mala sincronización de los recursos utilizados.

En la figura 4.53 se muestra el comportamiento del inventario en el centro de distribución.

Figura 4.53 Cantidad de mercancía en bodega, en relación con la hora del día.

Se observa bastante volumen de mercancía entre las ocho de la mañana y las siete de la noche, con un pico hacia las tres de la tarde.

En la figura 4.54 se muestra el tiempo de descargue de dos *vanes* (tractomulas) durante la simulación. La primera llevaba una carga de 500 unidades aproximadamente, y se demoraron en descargarla tres horas y media aproximadamente. La segunda tractomula llevaba más o menos la misma cantidad de unidades, y se demoraron descargándola una hora y media. Para el ingeniero Saavedra era importante conocer el porqué de la

diferencia tan grande en el de tiempo de descargue entre dos vehículos similares con cargas semejantes.

Darío analizó cuidadosamente la pendiente de la curva de descargue de la primera tractomula, y se dio cuenta de que, después de media hora de operación, había un cambio drástico en su comportamiento. Investigando con los encargados de la operación, sobre la posible causa de dicho cambio, encontró que éste había sido causado por el ingreso de personal de otro turno a esa hora (la tractomula estaba siendo descargada por seis operarios; después de media hora de operación se retiraron tres operarios, que habían terminado su turno de ocho horas). En la segunda tractomula se utilizaron seis operarios permanentes. Los resultados se pueden apreciar en la gráfica.

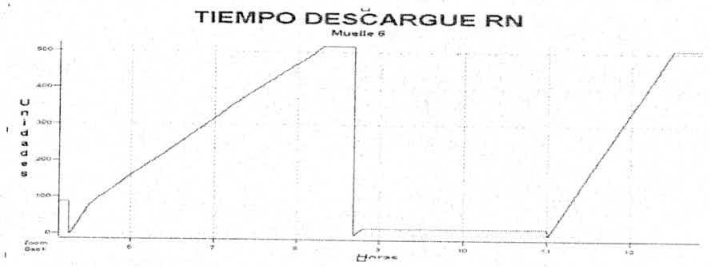


Figura 4.54 Tiempos de descargue de dos *vanes*.

Propuestas de mejoramiento

Utilización del recurso humano

En el modelo que elaboró Darío para simular la situación real del centro de distribución, *la hora de entrada de los auxiliares de distribución* se tomó como variable por sensibilizar. Estas personas normalmente eran contratadas por horas.

Partiendo de la *situación actual* del centro de distribución, y variando la hora de entrada de los auxiliares en la simulación, Darío encontró que la mejor opción era hacer que el personal entrara a trabajar a las 10 a.m., y no a las 6:00 a.m. como se venía haciendo. Con este cambio, aumentaría el número de unidades movilizadas por persona en un 8%, y se pasaría de mover 43 unidades por persona a 52 unidades por persona y, por supuesto, se reducirían los costos de operación.

Utilización incorrecta de los vehículos (*vanes*)

Los vanes se empleaban en la ruta nacional para llevar la mercancía de un centro de distribución de una ciudad al de otra. Los *vanes* se distribuían mal debido a que la asignación de rutas era aleatoria y no existía un método adecuado para hacerlo. Esto ocasionaba que el porcentaje de ocupación de los vehículos fuera diferente en el mismo periodo.

Darío simuló la operación de la ruta nacional buscando un método para lograr una mejor programación de los vehículos. A partir de dicha simulación, el ingeniero Saavedra mostró alternativas que no se habían podido experimentar en la realidad debido a los altos costos y al efecto de la resistencia al cambio de algunos encargados de la programación de vehículos. Con este modelo, Darío explicó a los gerentes de los demás centros de distribución por qué era más eficiente la nueva programación propuesta de los vehículos y los beneficios que ésta traería al implantarse ésta.

Gracias a la simulación de este proceso, ServiECI redujo el número de vehículos utilizados, cambió la forma de programarlos y comenzó a utilizar los vehículos de forma adecuada. Esto significó a la empresa un ahorro considerable en el costo de operación.

Conclusiones

La simulación con *ProModel*® es una herramienta que sirve para analizar y mejorar el proceso de una empresa de bienes o de servicios. Gracias a la simulación del ingeniero Saavedra, el centro de distribución de Bogotá mejoró sus procesos de cargue y descargue de los camiones, se detectó la causa de la baja productividad de los auxiliares de distribución y se mejoró la programación de los vehículos que transportan la mercancía en el territorio nacional. La simulación es un medio en el que se pueden hacer toda clase de supuestos físicos, económicos y humanos para tener el mejor o peor escenario que se pueda llegar a presentar en la realidad.

Nota: El modelo anterior, descrito de manera abreviada, así como los demás modelos aparecen en el CD que acompaña esta edición. Si usted se encuentra interesado en comunicarse con los autores lo puede hacer en el teléfono (571) 6762666 Ext. 278 de Bogotá, lblanco@escuelaing.edu.co, y con el editor en el correo *editor@escuelaing.edu.co*.

Índice de figuras

Figura 1.1	Pasos para realizar una simulación	7
Figura 1.2	Principales limitaciones en un estudio	9
Figura 1.3	Grado de afectación del tiempo de desarrollo causado por el nivel de detalle	11
Figura 1.4	Proyecto por etapas	13
Figura 1.5	Menú Archivos	16
Figura 1.6	Menú Información General	16
Figura 1.7	Menú Fusión o Mezcla	18
Figura 1.8	Menú para empaquetar	19
Figura 1.9	Menú para instalar modelo empaquetado	20
Figura 1.10	Menú de edición	20
Figura 1.11	Menú para construcción de modelos	21
Figura 1.12	Menú de gráficos	22
Figura 1.13	Ventanas del menú de sitios de trabajo	22
Figura 1.14	Menú de tiempos de parada	23
Figura 1.15	Menú de tiempos de parada por reloj	24
Figura 1.16	Menú de tiempo de parada por entradas	25
Figura 1.17	Menú de tiempo de parada por uso	26
Figura 1.18	Menú de tiempos de alistamiento	27
Figura 1.19	Menú para entidades	28
Figura 1.20	Ventanas de las redes de recorridos	29
Figura 1.21	Menú Interfaces	30
Figura 1.22	Ventana para colocar recursos	31
Figura 1.23	Ventana para especificaciones del recurso	32
Figura 1.24	Menú para definir el proceso	33
Figura 1.25	Ventana para definir la entrada al proceso	34
Figura 1.26	Ventana para definir la salida del proceso	34
Figura 1.27	Ventana para colocar instrucciones	35
Figura 1.28	Ventana para definir llegadas	35
Figura 1.29	Ventana para definir variables	36

Figura 1.30	Menú para definir condición de la simulación	38
Figura 2.1	Background de una industria manufacturera	42
Figura 2.2	Layout del modelo con colas para entrada y salida	46
Figura 2.3	Entidades del modelo	46
Figura 2.4	Instalaciones del modelo	46
Figura 2.5	Llegadas del modelo	47
Figura 2.6	Atributos del modelo	47
Figura 2.7	Variables del modelo	48
Figura 2.8	Layout con tres bandas de salida	50
Figura 2.9	Entidades del modelo	50
Figura 2.10	Gráficos de entidades	51
Figura 2.11	Instalaciones del modelo	51
Figura 2.12	Llegadas del modelo	52
Figura 2.13	Variables del modelo	52
Figura 2.14	Layout del modelo con tres entradas y tres salidas	54
Figura 2.15	Entidades del modelo	54
Figura 2.16	Instalaciones del modelo	55
Figura 2.17	Llegadas del modelo	55
Figura 2.18	Variables del modelo	55
Figura 2.19	Layout con cinco máquinas, una cola de entrada y otra de salida	57
Figura 2.20	Entidades del modelo	57
Figura 2.21	Instalaciones del modelo	58
Figura 2.22	Llegadas del modelo	60
Figura 2.23	Ciclo de llegadas del modelo	60
Figura 2.24	Tabla de tiempos del ciclo de llegadas del modelo	60
Figura 2.25	Variables del modelo	61
Figura 2.26	Layout del modelo	62
Figura 2.27	Entidad del modelo con dos gráficos	62
Figura 2.28	Instalaciones del modelo	63
Figura 2.29	Llegadas del modelo	63
Figura 2.30	Variables del modelo	64
Figura 2.31	Ventana para seleccionar las variables del modelo que se quieren analizar	65
Figura 2.32	A la izquierda, resultados sin variabilidad; a la derecha, resultados con variabilidad	66
Figura 2.33	Layout del modelo con dos rutas: una triangular y una recta	67
Figura 2.34	Layout con zona de descanso para los operarios	71
Figura 2.35	Entidad del modelo	71
Figura 2.36	Instalaciones del modelo	71
Figura 2.37	Llegadas del modelo	72
Figura 2.38	Variables del modelo	72
Figura 2.39	Redes del modelo	72
Figura 2.40	Recursos del modelo	72

Figura 2.41	Turno de ocho horas con dos descansos de veinte minutos	73
Figura 2.42	Asignación de turnos del modelo	74
Figura 2.43	Opción para definir el tiempo de la simulación del modelo	75
Figura 2.44	Layout del modelo en donde se destacan dos zonas de almacenamiento	76
Figura 2.45	Entidad del modelo con tres gráficos	76
Figura 2.46	Instalaciones del modelo	77
Figura 2.47	Llegadas del modelo	77
Figura 2.48	Atributos del modelo	77
Figura 2.49	Variables del modelo	78
Figura 2.50	Gráficos dinámicos de la simulación	79
Figura 2.51	Curva de ajuste de datos mediante Stat::Fit	81
Figura 2.52	Librería resultante de la fusión de dos existentes	81
Figura 2.53	Forma como se deben colocar las librerías; la nueva debe ir a la derecha	82
Figura 2.54	Ilustración de la forma como se copian dibujos en la librería	83
Figura 2.55	Importación de imágenes predeterminadas a la librería	83
Figura 2.56	Entidad del modelo con tres gráficos	84
Figura 2.57	Instalaciones del modelo	84
Figura 2.58	Llegadas del modelo	84
Figura 2.59	Variables del modelo	85
Figura 2.60	Resultados de la simulación del modelo	86
Figura 2.61	Layout del modelo con cuatro estaciones de trabajo	87
Figura 2.62	Entidad del modelo	87
Figura 2.63	Instalaciones del modelo	88
Figura 2.64	Ejemplo de definición de un macro	88
Figura 2.65	Llegadas del modelo	89
Figura 2.66	Variables del modelo	89
Figura 2.67	Definición de escenario	91
Figura 2.68	Layout del modelo A	91
Figura 2.69	Instalaciones del modelo A	92
Figura 2.70	Entidad del modelo A	92
Figura 2.71	Llegadas del modelo A	92
Figura 2.72	Layout del modelo B	93
Figura 2.73	Instalaciones del modelo B	93
Figura 2.74	Entidad del modelo B	93
Figura 2.75	Llegadas del modelo B	93
Figura 2.76	Layout conjunto de los modelos A y B	94
Figura 2.77	Simulación del modelo conjunto A y B	94
Figura 2.78	Ilustración del cargue y descargue de un pallet	95
Figura 2.79	Layout e instalaciones del modelo	95
Figura 2.80	Entidades del modelo	96
Figura 2.81	Llegadas al modelo	96

Figura 2.82 Atributos del modelo 96
Figura 2.83 Variables del modelo 97
Figura 2.84 Layout de un proceso de pasteurización de leche 98
Figura 2.85 Instalaciones del proceso de pasteurización 98
Figura 2.86 Figuras de fondo de los tanques y de la tubería sobre los que se colocan los medidores de nivel 99
Figura 2.87 Entidades del modelo 99
Figura 2.88 Llegadas al sistema 99
Figura 2.89 Ciclo de llegadas al sistema 100
Figura 2.90 Atributos del modelo 100
Figura 2.91 Variables del modelo 100
Figura 2.92 Redes del modelo 100
Figura 2.93 Trayectoria e interfaces de la Red 1 101
Figura 2.94 Trayectoria e interfaces de la Red 2 101
Figura 2.95 Trayectoria e interfaces de la Red 3 101
Figura 2.96 Trayectoria e interfaces de la Red 5 101
Figura 2.97 Recursos del modelo 101
Figura 3.1 Relación de la variabilidad con el *Throughput*, el *WIP* y el *Cycle time* 111
Figura 3.2 Distribución de la planta (layout) de Industrias ECI 111
Figura 3.3 Instalaciones del modelo 112
Figura 3.4 Entidades del sistema 112
Figura 3.5 Llegadas al sistema 113
Figura 3.6 Definición de la red *Desplazamientos* 113
Figura 3.7 Interfaces de la red de movimientos *Desplazamientos* 114
Figura 3.8 Recursos del modelo 114
Figura 3.9 Atributos del sistema 115
Figura 3.10 Variables del sistema 115
Figura 3.11 Subrutina *id* 116
Figura 3.12 Barra de herramientas de los reportes 121
Figura 3.13 Gráficas ProModel del *WIP*, *Throughput* y *Cycle Time* para cada una de las entidades *pieza*, *piñón* y para las dos, simultáneamente (*general*), de un sistema *Push*. Obsérvese que, en este sistema, el *WIP* se dispara 122
Figura 3.14 Utilización de instalaciones. Obsérvese que la operación cuello de botella es el *torno* 122
Figura 3.15 Distribución (*layout*) del cajero automático 126
Figura 3.16 Lógica de iniciación del programa 129
Figura 3.17 Paso del archivo de texto a Excel 130
Figura 3.18 Diagrama de barras Excel de la demanda diaria con tiempos discretos 131
Figura 3.19 Diagrama de pastel Excel del porcentaje de personas atendidas según la hora 132

Figura 3.20 Gráfica de ProModel que muestra el número de personas en cola en el cajero 132
Figura 3.21 Datos estadísticos de ProModel que muestran los clientes que abandonaron el sistema, debido a que la cola era muy larga 132
Figura 3.22 Gráfica Excel de la distribución de la llegada de clientes según la hora 133
Figura 3.23 Diagrama de barras Excel de la demanda diaria con tiempos aleatorios 135
Figura 3.24 Datos estadísticos de ProModel que muestran el número de clientes que abandonaron el sistema porque la cola era muy larga, con tiempos aleatorios 135
Figura 3.25 Número de personas en cola, en el modelo aleatorio 135
Figura 3.26 Porcentaje de ocupación del cajero de acuerdo con la hora, en el modelo aleatorio 136
Figura 3.27 Distribución (*layout*) del banco 139
Figura 3.28 Vistas: todo, cajeros, cuatro cajeros 142
Figura 3.29 Ruta de movimiento de los cajeros 143
Figura 3.30 Trayectorias e interfaces de la ruta Net 1 143
Figura 3.31 Especificaciones del recurso Res 1 144
Figura 3.32 Información de costos del recurso Res 1 147
Figura 3.33 Tiempos promedio en el sistema y número de clientes atendidos. Caso discreto 149
Figura 3.34 Clientes no atendidos. Tiempos de llegada y de servicio discretos 150
Figura 3.35 Costos de operación del banco con tiempos de llegada y de servicio discretos 151
Figura 3.36 Cuadro comparativo con tiempos de llegada y de servicio discretos 152
Figura 3.37 Tiempos promedios en el sistema, con tiempos de llegada y de servicio aleatorios 153
Figura 3.38 Clientes no atendidos, con tiempos de llegada y de servicio aleatorios 154
Figura 3.39 Costos de operación con tiempos de llegada y tiempos de servicio aleatorios 155
Figura 3.40 Cuadro comparativo con tiempos de llegada y de servicio aleatorios 156
Figura 3.41 Número de clientes atendidos por los cajeros para los diferentes tipos de demanda y porcentajes de atención, para el caso discreto 156
Figura 3.42 Número de clientes atendidos por los cajeros para los diferentes tipos de demanda y porcentajes de atención, para el caso aleatorio 157

Figura 3.43	Gráficas ProModel de llegada de clientes y de utilización de los recursos para el caso discreto	158
Figura 3.44	Gráficas ProModel de llegada de clientes y de utilización de los recursos para el caso aleatorio	159
Figura 3.45	Distribución (*layout*) del Sistema MRP	164
Figura 3.46	Entidades del modelo	165
Figura 3.47	Red desplazamientos	166
Figura 3.48	El recurso *Montacargas* tiene dos gráficos	167
Figura 3.49	Variables del modelo	168
Figura 3.50	Ventana de opciones para la simulación	171
Figura 3.51	Resultados de la simulación con MRP	172
Figura 3.52	Distribución (*layout*) del Sistema JIT o KANBAN	173
Figura 3.53	Instalaciones adicionales del modelo	174
Figura 3.54	Entidades del modelo	175
Figura 3.55	Red Ruta_dos del modelo	175
Figura 3.56	Variables, atributos, recursos y llegadas del modelo	177
Figura 3.57	Subrutinas del modelo	177
Figura 3.58	Tiempo de simulación	179
Figura 3.59	Resultados de la simulación JIT	180
Figura 3.60	Distribución (*layout*) del sistema CONWIP	180
Figura 3.61	Recursos del modelo	181
Figura 3.62	Llegadas al modelo	181
Figura 3.63	Resultados de la simulación con CONWIP	183
Figura 3.64	Distribución (*layout*) del sistema DRB	184
Figura 3.65	Recursos del modelo	184
Figura 3.66	Llegadas al sistema	185
Figura 3.67	Resultados del modelo DBR	186
Figura 4.1	*Layout* con tres alternativas para cargue y descargue de camiones	193
Figura 4.2	Instalaciones del modelo	194
Figura 4.3	Entidades del modelo	194
Figura 4.4	Red 1 del modelo	194
Figura 4.5	Red 2 del modelo	195
Figura 4.6	Red 3 del modelo	195
Figura 4.7	Red 5 del modelo	195
Figura 4.8	Recursos del modelo	196
Figura 4.9	Llegadas al sistema	197
Figura 4.10	Variables del modelo	197
Figura 4.11	Atributos del modelo	197
Figura 4.12	Subrutina mm del modelo	198
Figura 4.13	Macro para definir el número de personas dentro del *van*	199
Figura 4.14	Alternativas para descargue de camiones	202
Figura 4.15	Alternativas para descargue de camiones	203

Figura 4.16	Mapa digitalizado de la zona nororiental de Bogotá, con los nodos que se definen en las intersecciones	207
Figura 4.17	Puntos por donde viaja el furgón	209
Figura 4.18	Escogencia de la mejor ruta por el modelo	210
Figura 4.19	Direcciones de la mejor ruta sugerida por el modelo	210
Figura 4.20	Partes que integran el sistema logístico o cadena de abastecimientos	213
Figura 4.21	Actividades principales en el sistema logístico o cadena de abastecimientos	213
Figura 4.22	Papel integrador de la gerencia en el sistema logístico o cadena de abastecimientos	215
Figura 4.23	Modelo de simulación del sistema logístico o cadena de abastecimientos	216
Figura 4. 24	*Layout* del muelle de intercambio rápido	226
Figura 4. 25	Entidades que fluyen desde los proveedores hasta los puntos de venta	227
Figura 4.26	Llegadas al sistema	228
Figura 4.27	Atributos del modelo	228
Figura 4.28	Variables del modelo	228
Figura 4.29	Redes del modelo	228
Figura 4.30	Red fab-uno con sus interfaces	229
Figura 4.31	Red fab-dos con sus interfaces	229
Figura 4.32	Red fab-tres con sus interfaces	229
Figura 4.33	Red Distribución y sus interfaces	230
Figura 4.34	Red Distri_uno y sus interfaces	230
Figura 4.35	Red Distri_dos y sus interfaces	230
Figura 4.36	Red Distri_tres	230
Figura 4.37	Red Distri_cuatro	231
Figura 4.38	Recursos del modelo	231
Figura 4.39	Opciones de la simulación del modelo	234
Figura 4.40	Comportamiento de los tiempos de alistamiento, despacho y recibo	235
Figura 4.41	Procesos de recolección y distribución en el centro de distribución	239
Figura 4.42	Vista integral del *layout* del modelo. En la parte superior izquierda está el centro de llamadas; abajo, el muelle del centro de distribución y, a la izquierda, el mapa de rutas en la zona norte de Bogotá	240
Figura 4.43	*Layout* del muelle del centro de distribución de Bogotá. Abajo la zona en donde los operarios esperan la llegada de los vehículos y arriba, a la izquierda, el control de documentación	241

Figura 4.44 Algunas de las instalaciones del modelo correspondientes al muelle del centro de distribución 241
Figura 4.45 Algunas de las entidades consideradas en el modelo 242
Figura 4.46 Centro de control y de llamadas del sistema de información 242
Figura 4.47 Rutas de recolección y distribución en la zona norte de Bogotá 242
Figura 4.48 Vista integral del modelo en el momento de la simulación 243
Figura 4.49 Operación del muelle del centro de distribución durante la simulación 243
Figura 4.50 Operación del sistema de información y control durante la simulación 244
Figura 4.51 Resultados de la simulación. Los auxiliares permanecían en operación menos del cincuenta por ciento 245
Figura 4.52 Resultados de la simulación: tiempo de descargue para el proceso de distribución y tiempo de cargue para recolección 245
Figura 4.53 Cantidad de mercancía en bodega, en relación con la hora del día 246
Figura 4.54 Tiempos de descargue de dos *vanes* 247