

**IMPLEMENTACION DE TALLERES BASADOS EN EL SOFTWARE DE  
SIMULACIÓN FLEXSIM PARA LA ASIGNATURA TÉCNICAS MODERNAS DE  
OPTIMIZACIÓN**



**WILLIAM JESUS PINTO TRIANA**



**UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS  
ESCUELA DE ESTUDIOS INDUSTRIALES Y EMPRESARIALES  
BUCARAMANGA**

**2013**

**IMPLEMENTACION DE TALLERES BASADOS EN EL SOFTWARE DE  
SIMULACIÓN FLEXSIM PARA LA ASIGNATURA TÉCNICAS MODERNAS DE  
OPTIMIZACIÓN**

**WILLIAM JESÚS PINTO TRIANA**

**Trabajo de Grado para Optar al Título de  
Ingeniero Industrial**

**Director**

**EDWIN ALBERTO GARAVITO HERNÁNDEZ  
Ingeniero Industrial**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS  
ESCUELA DE ESTUDIOS INDUSTRIALES Y EMPRESARIALES  
BUCARAMANGA**

**2013**

## DEDICATORIA

*A Dios.*

*Porque todo lo que tengo se lo debo a Él y por las oportunidades que me brinda día a día. Por la realización de este trabajo y haberme dado la oportunidad de lograr mis objetivos, además de su infinita bondad, misericordia y amor.*

*A mis Padres.*

*Por su apoyo, sus sabios consejos pero sobre todo por su infinita paciencia.*

*A Ingrid Johanna.*

*Por su amor, su apoyo y la compañía que me ha brindado.*

*A mis Amigos.*

*Oscar y Juan por su colaboración durante el desarrollo del proyecto.*

*David y Andrea por su ayuda y preocupación.*

*Gracias por su amistad.*

*A todas las personas que me apoyaron incondicionalmente, me animaron a continuar adelante y que siempre creyeron en mí.*

*WILLIAM J. PINTO*

## AGRADECIMIENTOS

Al Ingeniero Industrial **Edwin Alberto Garavito Hernández**, Profesor de la Escuela de Estudios Industriales y Empresariales de la Universidad Industrial de Santander, director de este trabajo de grado, por su colaboración, paciencia y por la oportunidad brindada.

A **Oscar Fernando Arias** y **Juan Benedicto Peña**, Auxiliares de la sala de cómputo de la Escuela de Estudios Industriales y Empresariales, por su apoyo y colaboración durante el desarrollo del proyecto y ayuda con la licencia de Flexsim.

A la **Escuela de Estudios Industriales y Empresariales**, por el apoyo en el desarrollo del presente trabajo de grado.

# CONTENIDO

pág.

<b>INTRODUCCIÓN.....</b>	<b>19</b>
<b>1. GENERALIDADES.....</b>	<b>21</b>
1.1 DESCRIPCIÓN DE LA ASIGNATURA TÉCNICAS MODERNAS DE OPTIMIZACIÓN .....	21
1.2 PLANTEAMIENTO DEL PROBLEMA .....	22
1.3 JUSTIFICACIÓN .....	23
1.4 OBJETIVOS .....	23
1.4.1 Objetivo General .....	23
1.4.2 Objetivos Específicos.....	23
1.5 ALCANCE DEL TRABAJO .....	24
1.6 IMPACTO DEL PROYECTO DE GRADO .....	25
<b>2. MARCO TEORICO .....</b>	<b>26</b>
2.1 ¿QUÉ ES SIMULACIÓN? .....	26
2.2 TIPOS DE SISTEMAS .....	27
2.3 TIPOS DE MODELOS DE SIMULACIÓN .....	30
2.4 ¿POR QUÉ SIMULAR LOS PROCESOS? .....	33
2.5 VENTAJAS Y DESVENTAJAS DE LA SIMULACIÓN .....	33
2.6 APLICACIONES DE LA SIMULACIÓN .....	36
2.7 PASOS PARA REALIZAR UN ESTUDIO DE SIMULACIÓN.....	37
2.7.1 Definición del sistema bajo estudio .....	37
2.7.2 Generación del modelo de simulación base .....	38
2.7.3 Recolección y análisis de datos .....	38
2.7.4 Generación del modelo preliminar .....	38
2.7.5 Verificación del modelo .....	38
2.7.6 Validación del modelo.....	39

2.7.7	Generación del modelo final.....	39
2.7.8	Determinación de los escenarios para el análisis .....	39
2.7.9	Análisis de sensibilidad .....	39
2.7.10	Documentación del modelo.....	39
2.8	PROYECTOS DE SIMULACIÓN .....	41
2.9	SIMULACIÓN MONTECARLO.....	42
2.10	GENERACIÓN DE NÚMEROS ALEATORIOS.....	43
2.11	GENERACIÓN DE VARIABLES ALEATORIAS .....	44
2.11.1	Método de la Transformada Inversa.....	46
2.11.2	Método de Aceptación y Rechazo.....	48
2.11.3	Método de Convolución.....	50
2.12	TÉCNICAS DE REDUCCIÓN DE VARIANZA .....	51
2.12.1	Variables antitéticas.....	52
2.12.2	Variables de control .....	52
2.12.3	Números aleatorios comunes .....	54
<b>3.</b>	<b>SOFTWARE DE SIMULACIÓN.....</b>	<b>55</b>
3.1	COMPONENTES DE UN SOFTWARE DE SIMULACIÓN.....	56
3.2	SOFTWARE EMPLEADO EN LA ACADEMIA E INDUSTRIA.....	58
3.2.1	FLEXSIM .....	59
3.2.2	PROMODEL.....	63
3.2.3	ARENA.....	66
3.2.4	WITNESS .....	69
3.2.5	AUTOMOD .....	71
3.2.6	EXTENDSIM.....	72
3.2.7	SIMUL8.....	72
<b>4.</b>	<b>INSTALACIÓN Y CONFIGURACIÓN DE FLEXSIM.....</b>	<b>73</b>
4.1	ACCESO REMOTO CON VPN .....	74

<b>5. METODOLOGÍA DE ENSEÑANZA-APRENDIZAJE .....</b>	<b>76</b>
5.1 LAS TICS EN LOS PROCESOS DE ENSEÑANZA Y APRENDIZAJE .....	76
5.2 ESTRATEGIAS PARA LA ENSEÑANZA .....	78
5.3 ESTRATEGIA DIDÁCTICA.....	81
5.3.1 Diseño Instruccional del Objeto de Aprendizaje .....	82
5.4 ESTRATEGIAS PARA EL APRENDIZAJE .....	82
<b>6. INSTRUCTIVO DE APRENDIZAJE DEL SOFTWARE DE SIMULACIÓN FLEXSIM .</b>	<b>85</b>
6.4 DISEÑO.....	87
6.4.1 ¿A quién va dirigido este Instructivo de Aprendizaje? .....	87
<b>7. TALLERES PRÁCTICOS DE SIMULACIÓN CON FLEXSIM .....</b>	<b>89</b>
7.1 PLANEACIÓN Y DESARROLLO.....	89
7.2 TALLERES DESARROLLADOS EN LA ASIGNATURA.....	91
7.3 EVALUACIÓN DE LA PRUEBA PILOTO .....	99
7.3.1 Presentación .....	102
7.3.2 Lenguaje .....	105
7.3.3 Cobertura y Profundidad .....	107
7.3.4 Utilidad.....	110
7.4 CONCLUSIONES DE LA EVALUACIÓN DE LA PRUEBA PILOTO .....	113
<b>8. CONCLUSIONES.....</b>	<b>114</b>
<b>9. RECOMENDACIONES.....</b>	<b>116</b>
<b>BIBLIOGRAFÍA .....</b>	<b>118</b>

## LISTA DE TABLAS

Tabla 1. Software Flexsim. ....	59
Tabla 2. Software ProModel .....	63
Tabla 3. Arena .....	66
Tabla 4. Witness .....	70
Tabla 7. Taller Introductorio .....	91
Tabla 8. Taller No. 1 .....	93
Tabla 9. Taller No. 2 .....	94
Tabla 10. Taller No. 3 .....	95
Tabla 11. Taller No. 4 .....	96
Tabla 12. Taller No. 5 .....	97
Tabla 13. Taller No. 6 .....	98
Tabla 14. Resultado general del componente Presentación.....	105
Tabla 15. Resultado general del componente Lenguaje .....	107
Tabla 16. Resultado general del componente Cobertura y Profundidad.....	109
Tabla 17. Resultado general del componente Utilidad.....	112

## LISTA DE FIGURAS

Figura 1. Sistema Continuo .....	28
Figura 2. Sistema Discreto.....	29
Figura 3. Tipos de Modelos .....	30
Figura 4. Pasos de una simulación .....	40
Figura 5. Ejemplo de Modelo de Simulación en Flexsim. ....	62
Figura 6. Ejemplo de Modelo de Simulación en ProModel. ....	65
Figura 7. Gráficos estadísticos en ProModel. ....	65
Figura 8. Documentos de WCS haciendo referencia a Arena. ....	68
Figura 9. Modelo de Simulación en Arena. ....	69
Figura 10. Modelo de distribución de planta en Witness.....	70
Figura 11. Centro de Distribución en AutoMod. ....	71
Figura 12. VPN con el software Hamachi .....	75
Figura 13. Estrategia de Enseñanza - Aprendizaje.....	76
Figura 14. Estrategias de enseñanza involucradas en los talleres .....	81
Figura 15. Características del instructivo de aprendizaje.....	87
Figura 16. Diseño de los talleres.....	90
Figura 17. Componentes de la evaluación de la prueba piloto .....	99
Figura 18. Componente de la encuesta: Presentación .....	100
Figura 19. Componente de la encuesta: Lenguaje .....	100
Figura 20. Componente de la encuesta: Cobertura y Profundidad .....	101
Figura 21. Componente de la encuesta: Utilidad .....	101
Figura 22. Resultados pregunta 1 del componente Presentación.....	102
Figura 23. Resultados pregunta 2 del componente Presentación.....	103
Figura 24. Resultados pregunta 3 del componente Presentación.....	103
Figura 25. Resultados pregunta 4 del componente Presentación.....	104
Figura 26. Resultados pregunta 5 del componente Presentación.....	104
Figura 27. Resultados pregunta 1 del componente Lenguaje.....	105

Figura 28. Resultados pregunta 2 del componente Lenguaje.....	106
Figura 29. Resultados pregunta 3 del componente Lenguaje.....	106
Figura 30. Resultados pregunta 1 del componente Cobertura y Profundidad .....	107
Figura 31. Resultados pregunta 2 del componente Cobertura y Profundidad .....	108
Figura 32. Resultados pregunta 3 del componente Cobertura y Profundidad .....	108
Figura 33. Resultados pregunta 4 del componente Cobertura y Profundidad .....	109
Figura 34. Resultados pregunta 1 del componente Utilidad .....	110
Figura 35. Resultados pregunta 2 del componente Utilidad .....	110
Figura 36. Resultados pregunta 3 del componente Utilidad .....	111
Figura 37. Resultados pregunta 4 del componente Utilidad .....	111
Figura 38. Resultados pregunta 5 del componente Utilidad .....	112

## LISTA DE ANEXOS

ANEXO A. Instructivo de Aprendizaje del Software de Simulación Flexsim .....	121
ANEXO B. Talleres Prácticos con el Software de Simulación Flexsim .....	254
ANEXO C. Encuesta de Evaluación de los Talleres .....	384

## RESUMEN

### **TÍTULO:**

IMPLEMENTACION DE TALLERES BASADOS EN EL SOFTWARE DE SIMULACIÓN FLEXSIM PARA LA ASIGNATURA TÉCNICAS MODERNAS DE OPTIMIZACIÓN.\*

### **AUTOR:**

WILLIAM JESÚS PINTO TRIANA\*\*

### **PALABRAS CLAVES:**

Simulación, Flexsim, Evento Discreto, Aprendizaje Colaborativo.

### **DESCRIPCIÓN:**

La Escuela de Estudios Industriales y Empresariales de la Universidad Industrial de Santander, ha fomentado en los últimos años el desarrollo de herramientas pedagógicas a través de las Tecnologías de Información y Comunicación (TICS), que permitan a los estudiantes mejorar las competencias en su formación profesional y académica, desarrollando habilidades en el manejo de software especializado para estar a la vanguardia de las mejores herramientas disponibles en el mercado para diseñar, modelar y simular procesos productivos y logísticos, logrando de esta manera una mayor competitividad en el mundo laboral y por consiguiente más oportunidades. Es por ello que este trabajo de grado surge al identificar la necesidad específica de implementar una metodología de enseñanza – aprendizaje en el diseño de las prácticas experimentales de la asignatura Técnicas Modernas de Optimización.

Finalmente la herramienta pedagógica consta de un instructivo de aprendizaje, un manual con seis practicas con elementos diferenciadores para cada una de ellas y que abordan la temática establecida, y que están enfocadas a brindar soporte a los usuarios del software de simulación Flexsim, donde se establecen las pautas para el satisfactorio desarrollo de las mismas; un banco de videos y modelos de guía en Flexsim para el uso de los estudiantes.

---

\* Proyecto de grado.

\*\* Facultad de Ingenierías Físico-Mecánicas, Escuela de Estudios Industriales y Empresariales.  
Director: Prof. Edwin Alberto Garavito Hernández.

## ABSTRACT

**TITLE:**

IMPLEMENTATION OF WORKSHOPS BASED ON THE SIMULATION SOFTWARE FLEXSIM FOR THE SUBJECT MODERN TECHNIQUES OF OPTIMIZATION.\*

**AUTHOR:**

WILLIAM JESÚS PINTO TRIANA\*\*

**KEYWORDS:**

Simulation, Flexsim, Discrete-Event, Collaborative Learning.

**DESCRIPTION:**

The School of Industrial and Enterprise Studies from the Universidad Industrial de Santander, has fostered in recent years the development of educational tools through Information and Communication Technologies (ICT), which allow students to improve skills in your professional and academic training, developing skills in handling specialized software to be at the forefront of the best tools available in the market to design, model and simulate production processes and logistics, thus achieving way greater competitiveness in the workplace and therefore more opportunities. That is why this paper grade arises to identify the specific need to implement a methodology of teaching - learning in the design of experimental practices subject Modern Techniques of Optimization.

Finally pedagogical tool consists of an instructive learning, a manual with six practices with differentiators for each that address the topic set, which are aimed at providing support to users of Flexsim simulation software, which sets guidelines for the successful development of the same, a bank of videos and models Flexsim guide for student use.

---

\* Degree Project.

\*\* Faculty of Physical-Mechanical Engineering. School of Industrial and Enterprise Studies.  
Director: Prof. Edwin Alberto Garavito Hernández.

## GLOSARIO

- ❖ **ADD-ON:** llamado también agregado o plugin, es una extensión o programa que solo funciona anexo a otra y que sirve para incrementar o complementar sus funcionalidades. Tiene funciones más extensas que el plugin. Abarca plugins, extensiones, temas para las aplicaciones diseñadas.
  
- ❖ **APLICACIÓN INFORMÁTICA:** es un tipo de programa informático diseñado como herramienta para permitir a un usuario realizar uno o diversos tipos de trabajos.
  
- ❖ **AVA:** Ambiente Virtual de Aprendizaje, o *Virtual Learning Environment (VLE)*. Es un sistema de software diseñado para facilitar a profesores la gestión de cursos virtuales para sus estudiantes, especialmente ayudándolos en la administración y desarrollo del curso.
  
- ❖ **EEIE:** Escuela de Estudios Industriales y Empresariales.
  
- ❖ **ESTILOS DE APRENDIZAJE:** conjunto de características psicológicas y cognitivas que expresa una persona cuando aprende algo, son las distintas maneras en que un individuo puede aprender. Se encuentra muy relacionado con las estrategias de aprendizaje
  
- ❖ **ESTRATEGIAS DE APRENDIZAJE:** son los procedimientos que pone en acción un individuo cuando su meta es aprender algo.
  
- ❖ **ESTRATEGIAS DE ENSEÑANZA:** son los procedimientos que emplea un docente, o tutor para promover el aprendizaje.

- ❖ **EVA:** Entorno Virtual de Aprendizaje. Sinónimo de AVA.
- ❖ **INSTRUCTIVO DE APRENDIZAJE:** guía o manual que orienta a un individuo hacia adquirir nuevo conocimiento.
- ❖ **LAG:** retraso prolongado en una telecomunicación que dificulta la interacción entre los usuarios.
- ❖ **LAN:** Red de área local, del inglés *Local Area Network*. Es una interconexión de varios computadores y periféricos.
- ❖ **LAYOUT:** Esquema de distribución de los elementos en el área de operación.
- ❖ **LMS:** Sistema de Gestión del Aprendizaje, del inglés *Learning Management System*. Es una plataforma de software alojada en un servidor web y accesible desde Internet, que se emplea para administrar, distribuir y controlar las actividades de formación no presencial (o aprendizaje electrónico) de una institución u organización.
- ❖ **MODELO:** es una representación de un sistema con el fin de estudiarlo.
- ❖ **OBJETO DE APRENDIZAJE:** conjunto de recursos digitales estructurados utilizados durante un proceso de aprendizaje mediado por la tecnología, como lo son los contenidos multimedia, contenidos didácticos, herramientas de software, etc.
- ❖ **PLUG-IN:** llamado también complemento o plugin, es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica. Es un módulo de un programa.

- ❖ **SISTEMA:** conjunto de elementos que se interrelacionan para funcionar como un todo con unas fronteras claras y definidas para el logro de un objetivo, por ejemplo: el sistema de atención de clientes en un banco, el sistema de inventarios de una empresa o el sistema de atención en la sala de emergencia de un hospital.
  
- ❖ **TIC:** Tecnologías de la Información y la Comunicación. Conjunto de recursos, procedimientos y técnicas usadas en el procesamiento, almacenamiento y transmisión de información.
  
- ❖ **TMO:** Técnicas Modernas de Optimización. Electiva Técnica Profesional del plan de estudios de Ingeniería Industrial.
  
- ❖ **VPN:** Red privada virtual, del inglés *Virtual Private Network*. Permite una extensión segura de una red local LAN para su conexión remotamente.

## INTRODUCCIÓN

La simulación es una representación, una abstracción que imita el comportamiento de un sistema a través de un modelo que representa una realidad, y una poderosa herramienta en la toma de decisiones que permite aumentar la eficiencia y desempeño de una organización, así como reducir costos. Ha sido utilizada desde hace mucho tiempo gracias al surgimiento de la investigación operacional, pero ha cobrado gran importancia en las dos últimas décadas debido a la disponibilidad de los computadores personales y el desarrollo de nuevos lenguajes de programación, además de los altos costos de experimentar y hacer ensayos en la realidad, siendo una de las técnicas de mayor impacto en la actualidad gracias a los avances computacionales presentes.

Flexsim es el Software de simulación más innovador que existe hoy en día en el mercado para modelar, simular y visualizar sistemas y procesos en empresas manufactureras y organizaciones de servicio.

En Colombia, universidades reconocidas como la Universidad de Antioquia, la Universidad Distrital, la Universidad de los Andes, la Universidad Javeriana, la Universidad Nacional, la Universidad de la Sabana, la Escuela Colombiana de Ingeniería Julio Garavito, Universidad Tecnológica de Bolívar, la UPTC Sogamoso y la UPTC Duitama, entre otras, están utilizando Flexsim como software de simulación<sup>1</sup>. A nivel Internacional cuenta con usuarios como: Georgia Tech, MIT y Northwestern University entre otras, además de empresas reconocidas mundialmente como Coca-Cola, Whirlpool, NASA, General Mills, DuPont,

---

<sup>1</sup> Tomado de: Conozca las ventajas de Flexsim en Educación. Recuperado el 10 de agosto de 2011 en <http://www.vaticgroup.com/unlimitpages.asp?id=133>

Goodyear, Puerto de Cartagena, Siemens, Maersk, Norfolk Southern, Pfizer, Avon, Volkswagen, Ford, Mattel, Michelin, FedEx, Gillette y muchas otras.<sup>2</sup>

En la actualidad es necesario que las Universidades se adapten a las nuevas tecnologías y desarrollos disponibles, en aras de implementar estas mejoras en sus planes académicos. Por lo cual este proyecto de grado surge como respuesta a la necesidad que tienen los estudiantes de Ingeniería Industrial, en particular, de una herramienta metodológica que les permita acceder a información, eminentemente práctica, para el manejo del software de simulación Flexsim.

El presente documento incluye siete capítulos principales que describen el trabajo realizado. El primero de ellos cubre las generalidades del proyecto de grado, correspondientes a una breve descripción de la asignatura, planteamiento del problema, objetivo general y objetivos específicos, alcance e impacto.

Posteriormente, en el segundo capítulo, se realiza la revisión bibliográfica correspondiente a simulación, ventajas, sus aplicaciones, tipos de sistemas y modelos y generación de variables aleatorias.

Después de presentar las generalidades del proyecto de grado, la revisión bibliográfica correspondiente y el software empleado en la academia e industria, en los tres capítulos siguientes se describen las metodologías de enseñanza-aprendizaje, el diseño del instructivo de aprendizaje de Flexsim y los talleres implementados junto con la evaluación de la prueba piloto.

Por último se presentan las conclusiones y recomendaciones pertinentes.

---

<sup>2</sup> Tomado de: Flexsim Simulation Software – Some of our clients. Recuperado el 10 de febrero de 2011 en <http://www.flexsim.com/company/clients/>

## **1. GENERALIDADES**

### **1.1 DESCRIPCIÓN DE LA ASIGNATURA TÉCNICAS MODERNAS DE OPTIMIZACIÓN**

El proyecto se desarrollará en la Escuela de Estudios Industriales y Empresariales de la Universidad Industrial de Santander, específicamente para la asignatura Técnicas Modernas de Optimización, la cual es una electiva técnica profesional de la carrera de Ingeniería Industrial.

La creación de la asignatura Técnicas Moderas de Optimización nace de la necesidad de desarrollar competencias en los estudiantes que permitan analizar, modelar, experimentar sistemas productivos y de servicios de una organización, reales o hipotéticos a través de la simulación de eventos discretos y continuos con el fin de conocerlos con claridad y/o mejorar su funcionamiento.

#### **PROPÓSITOS DEL CURSO:**

- ❖ Analizar los conceptos fundamentales de la simulación de sistemas y su aplicación en la solución de modelos de optimización, sistemas de colas e inventarios y el análisis de sistemas dinámicos.
- ❖ Aprender las principales técnicas de simulación en lo referente a Simulación de sucesos discretos, simulación por intervalo y optimización.
- ❖ Utilizar los lenguajes de programación (MATLAB), simulación (SIMAN) y software de simulación (ARENA, PROMODEL, FLEXSIM), para el estudio y análisis de comportamientos de sistemas productivos y de servicios.

- ❖ Conocer las herramientas estadísticas para el análisis de los resultados de los métodos de simulación.
- ❖ Conocer las técnicas existentes para mejorar el procedimiento de simulación y mejorar la confiabilidad de los resultados.

## **1.2 PLANTEAMIENTO DEL PROBLEMA**

La Escuela de Estudios Industriales y Empresariales consciente de la necesidad de estar a la vanguardia de técnicas y herramientas utilizadas hoy en día tanto en la industria como en la academia, adquirió el software de simulación Flexsim para formar profesionales competentes en dicha temática.

La Escuela no cuenta con un laboratorio de cómputo con talleres formales para la enseñanza de esta herramienta, por lo tanto surge la necesidad de contar con un planteamiento temático y estratégico que favorezca el proceso de enseñanza y aprendizaje de Flexsim, con lo cual es necesario desarrollar una metodología en la instrucción de este software, proponiéndose la elaboración de un manual, material pedagógico y el diseño de talleres prácticos para implementar en un laboratorio de cómputo adecuado para tal fin dentro del programa de la asignatura Técnicas Modernas de Optimización, asegurando de este modo el proceso de formación de los estudiantes, permitiéndoles adquirir un conocimiento significativo sobre esta importante herramienta de simulación.

Adicionalmente se evidencian falencias de los estudiantes de Ingeniería Industrial en el uso de herramientas computacionales y software especializado para el proceso de toma de decisiones en entornos complejos como la industria manufacturera y de servicios, herramientas de apoyo que son de gran importancia

para un ingeniero en un mundo dinámico donde es incuestionable la importancia de la tecnología en todos los ámbitos sociales.

### **1.3 JUSTIFICACIÓN**

Este proyecto de grado servirá como base para el desarrollo de futuros proyectos de investigación, prácticas empresariales y actividades de extensión que involucren la simulación con Flexsim en la Escuela de Estudios Industriales y Empresariales.

### **1.4 OBJETIVOS**

#### **1.4.1 Objetivo General**

Diseñar, documentar e implementar talleres prácticos basados en el software de simulación Flexsim para la asignatura Técnicas Modernas de Optimización.

#### **1.4.2 Objetivos Específicos**

- ❖ Realizar una revisión bibliográfica sobre Simulación de Procesos y el software empleado en la academia y en la industria.
- ❖ Definir y plantear una metodología para la enseñanza y utilización de Flexsim.
- ❖ Diseñar 6 talleres prácticos para la formación de los estudiantes en el manejo del software de simulación Flexsim.

- ❖ Elaborar instructivo para el aprendizaje de Flexsim y el desarrollo de los talleres diseñados.
- ❖ Implementar 3 de los 6 talleres diseñados para validar su efectividad en el proceso de aprendizaje y dominio logrado por los estudiantes, a través de una prueba piloto.
- ❖ Documentar los resultados obtenidos.

## **1.5 ALCANCE DEL TRABAJO**

Los productos a obtener con la realización de este trabajo de grado son:

- ❖ Documento que incluya el marco teórico de Simulación de Procesos, con sus revisiones bibliográficas.
- ❖ Metodología a utilizar para el proceso de enseñanza y aprendizaje de Flexsim.
- ❖ Instalación y configuración de licencias concurrentes de Flexsim en el laboratorio de la asignatura, que incluya todo lo necesario para el desarrollo de los talleres diseñados.
- ❖ Documento de instrucciones de uso del software de simulación Flexsim, que incluya los talleres diseñados.
- ❖ Instrumento de evaluación de la prueba piloto y conclusiones.

## **1.6 IMPACTO DEL PROYECTO DE GRADO**

Este trabajo de grado sitúa a la Universidad Industrial de Santander a la vanguardia de la utilización de software especializado para la simulación de eventos discretos, continuos y mixtos, y aporta a los estudiantes de Ingeniería Industrial la formación integral de competencias y habilidades en el diseño, modelado y simulación de procesos productivos y logísticos, mejorando así su competitividad y el aprendizaje de una herramienta para la toma de decisiones y análisis de procesos, de esta forma el profesional de la UIS tendrá un mayor campo de acción laboral y por consiguiente más oportunidades.

## 2. MARCO TEORICO

### 2.1 ¿QUÉ ES SIMULACIÓN?

La simulación es una representación ficticia de la realidad, es una técnica que permite imitar (o simular) en un computador el comportamiento de un sistema físico o teórico, para estimar las medidas de desempeño del sistema modelado. “Cuanto mejor sea el grado de aproximación de la simulación a la realidad, mayor será su utilidad”<sup>3</sup>. Permite predecir y comprender cierto aspecto del comportamiento del sistema y evaluar de esta manera diferentes tipos de estrategias para su mejor operación. El requisito previo de cualquier simulación es un buen conocimiento del sistema real.

Es una de las técnicas más ampliamente usadas en Investigación de Operaciones y Ciencias de la Administración.

“El uso de la simulación como metodología de trabajo es una actividad muy antigua, y podría decirse que inherente al proceso de aprendizaje del ser humano... Para poder comprender la realidad y toda la complejidad que un sistema puede conllevar, ha sido necesario construir artificialmente objetos y experimentar con ellos dinámicamente antes de interactuar con el sistema real”<sup>4</sup>.

En particular, se usan computadores para imitar comportamientos del sistema, evaluando numéricamente un modelo del mismo. Estas evaluaciones numéricas son las que permiten generar las historias artificiales que no son más que experimentos. Se simula para explicar, entender o mejorar el sistema.

---

<sup>3</sup> Ver: Blanco, 2001, p. 3.

<sup>4</sup> Tomado de: Guasch, Piera, Casanovas, Figueras, 2005, p. 1.

La simulación es una de las 10 principales estrategias tecnológicas nombradas por Gartner Newsroom<sup>5</sup> (Sala de prensa de Gartner Inc.) ya que permite ver el futuro prediciendo lo que puede pasar.

Gartner Inc. es una empresa consultora y de investigación de las tecnologías de la información con sede en Stamford, Connecticut, Estados Unidos. Hasta 2001 era conocida como Gartner Group.

## 2.2 TIPOS DE SISTEMAS

Un sistema puede definirse como un conjunto de objetos o entidades que interactúan entre sí para alcanzar el logro de cierto objetivo.

Entre los componentes de un sistema se encuentran:

- ✓ Entidad: es un objeto de interés para el sistema.
- ✓ Atributo: es una propiedad de una entidad.
- ✓ Actividad: representa un período de tiempo de determinada duración.
- ✓ Recurso: Dispositivo necesario para llevar a cabo una operación.
- ✓ Estado de un sistema: conjunto mínimo de variables necesarias para caracterizar o describir todos aquellos aspectos de interés del sistema en un cierto instante de tiempo. Estas variables se denominan variables de estado.

---

<sup>5</sup> Consultado en: Gartner Identifies the Top 10 Strategic Technologies for 2010. Recuperado el 15 de septiembre de 2011 en: <http://www.gartner.com/it/page.jsp?id=1210613>

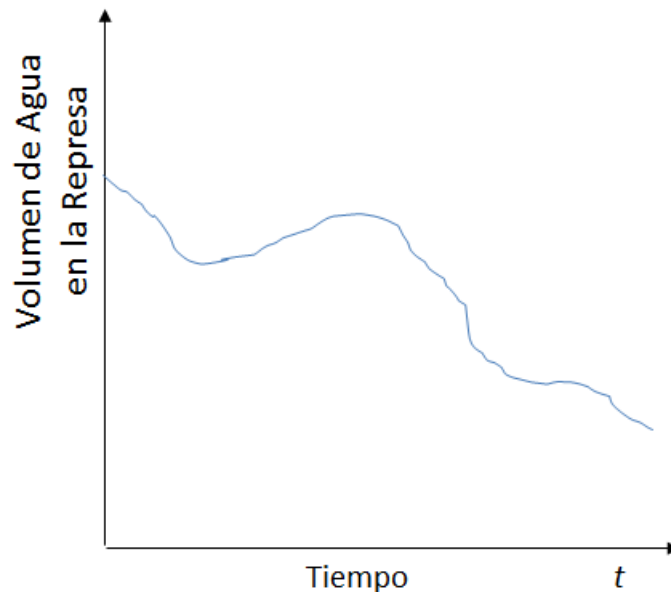
- ✓ Evento: es una ocurrencia instantánea que puede cambiar el estado de un sistema.

De acuerdo a lo anterior se puede afirmar que una actividad se encuentra comprendida entre dos eventos.

Según Guasch, atendiendo a la relación entre la evolución de las propiedades de interés y la variable independiente tiempo, los sistemas se clasifican en<sup>6</sup> :

- ❖ **Sistemas Continuos:** es aquel en el cual las variables del estado del sistema cambian continuamente en el tiempo. Ejemplo: el volumen de agua de una represa.

Figura 1. Sistema Continuo



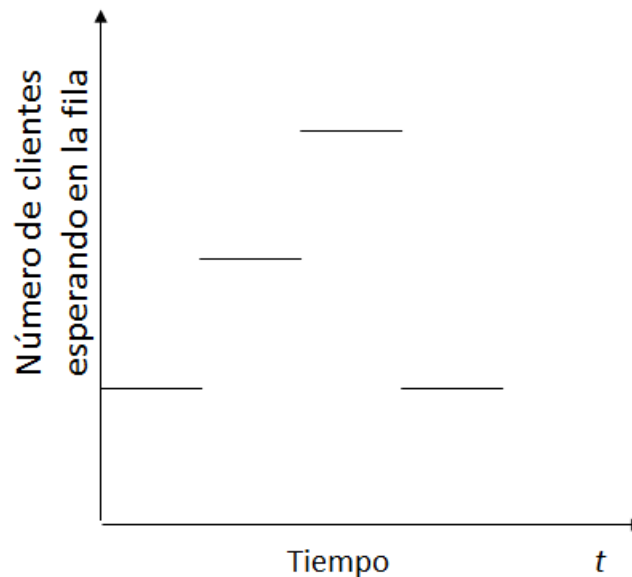
Fuente: BANKS, et al. (2001), pág. 12.

---

<sup>6</sup> Ver: Guasch, et al. 2005, p. 5.

- ❖ **Sistemas Discretos.** Se caracterizan porque las propiedades de interés del sistema cambian únicamente en un cierto instante o secuencia de instantes (puntos discretos en el tiempo), y permanecen constantes el resto del tiempo. La secuencia de instantes en los cuales el estado del sistema puede presentar un cambio, obedece normalmente a un patrón periódico.

Figura 2. Sistema Discreto



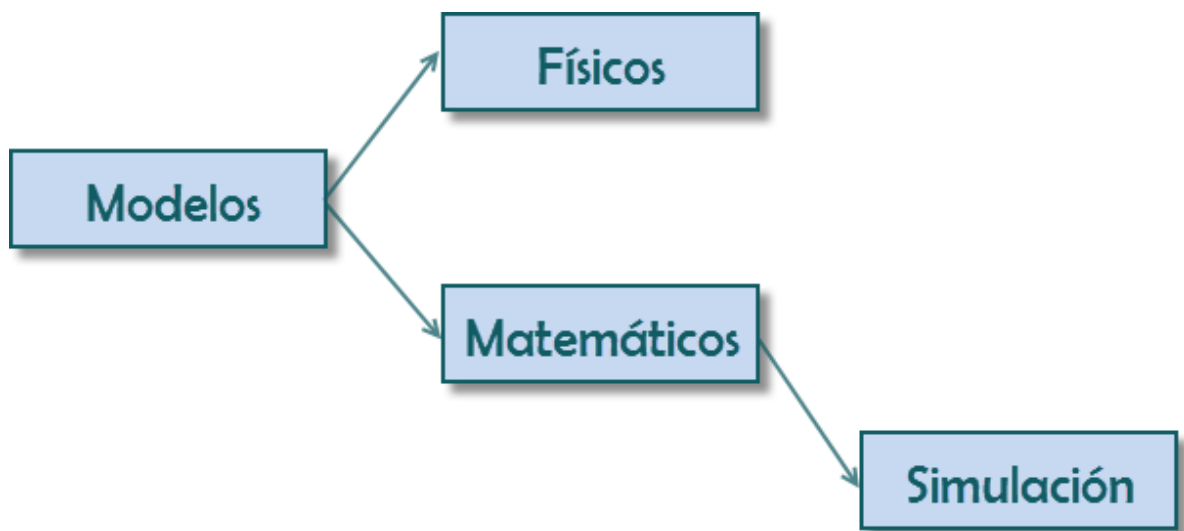
Fuente: BANKS, et al. (2001), pág. 12.

- ❖ **Sistemas orientados a eventos discretos.** Al igual que los sistemas discretos, se caracterizan porque las propiedades de interés del sistema cambian únicamente en una secuencia de instantes de tiempo permaneciendo constantes el resto del tiempo. La secuencia de instantes en los cuales el estado del sistema puede presentar un cambio, obedece a un patrón aleatorio.
- ❖ **Sistemas combinados.** Aquellos que combinan subsistemas que siguen filosofías continuas o discretas, respectivamente. Es el caso de los sistemas que poseen componentes que deben ser necesariamente modelados según alguno de dichos enfoques específicos.

### 2.3 TIPOS DE MODELOS DE SIMULACIÓN

**Modelo:** es una representación del sistema con el propósito de estudiarlo. La descripción de las características de interés de un sistema se conoce como *modelo del sistema*, y el proceso de abstracción para obtener esta descripción se conoce como *modelado*.

Figura 3. Tipos de Modelos



Fuente: Adaptado de GUASCH, et al. (2005), pág. 4.

Existen varios tipos de modelos para representar los sistemas objeto de estudio, como: modelos físicos, mentales, simbólicos, matemáticos.

Se utilizan modelos simbólicos matemáticos como herramienta para representar las dinámicas de interés de cualquier sistema en un entorno de simulación digital. Estos modelos de simulación suelen clasificarse de la siguiente forma<sup>7</sup>:

---

<sup>7</sup> Ibíd.

### ❖ **Modelos Estáticos frente a Modelos Dinámicos**

Los Modelos Estáticos: suelen utilizarse para representar el sistema en un cierto instante de tiempo; y en su formulación no se considera el avance del tiempo. Frecuentemente involucra muestras variables para generar un resultado estadístico, generalmente se conoce como la Simulación de Monte Carlo. Por ejemplo en el ámbito financiero, este tipo de simulación es usada para seleccionar un portafolio de acciones o bonos.

Los Modelos Dinámicos: permiten deducir cómo las variables de interés del sistema en estudio evolucionan con el tiempo. Tienen una o más variables aleatorias como entradas. La simulación dinámica es muy utilizada para el análisis de sistemas de manufactura y servicios, por ejemplo la simulación en una sala de emergencias de un hospital durante una jornada de trabajo habitual.

### ❖ **Modelos Deterministas respecto a Modelos Estocásticos**

Un modelo se denomina Determinista si su nuevo estado puede ser completamente definido a partir del estado previo y de sus entradas. Es decir, ofrece un único conjunto de valores de salida para un conjunto de entradas conocidas. Representa el sistema en un punto particular en el tiempo y no posee componentes de entrada de tipo aleatorio.

Los resultados obtenidos en una simulación de este tipo siempre van a ser los mismos independientemente del número de veces que la simulación sea corrida. Por ejemplo las llegadas a un consultorio médico, a diferencia de la sala de emergencias, pueden ser de tipo determinístico si cada uno de los pacientes llega a la hora determinada de su cita.

Los Modelos Estocásticos requieren de una o más variables aleatorias para formalizar las dinámicas de interés. En consecuencia, el modelo no genera un único conjunto de salidas cuando es utilizado para realizar un experimento, sino que los resultados son utilizados para estimar el comportamiento real del sistema. Representan el sistema a través del tiempo. Por ejemplo dentro del funcionamiento de la sala de emergencias de un hospital, pueden ser consideradas como variables estocásticas: el intervalo de tiempo de llegada de pacientes o el tiempo en el cual cada paciente es atendido por el médico dependiendo de la gravedad de su situación.

#### ❖ **Modelos Continuos frente a Modelos Discretos**

Los Modelos Continuos se caracterizan por representar la evolución de las variables de interés de forma continua. En general suelen utilizarse ecuaciones diferenciales ordinarias si se considera simplemente la evolución de una propiedad respecto al tiempo, o bien ecuaciones en derivadas parciales si se considera también la evolución respecto a otras variables adicionales. Un ejemplo de modelo continuo podría verse reflejado en determinar cómo fluye un líquido dentro de una tubería.

De modo análogo a la definición de los modelos continuos, los Modelos Discretos se caracterizan por representar la evolución de las variables de interés de forma discreta.

Según la clasificación anterior, el tipo de simulación en el cual se ha de centrar el desarrollo de este proyecto es: Simulación de eventos discretos dinámicos, que incluye variables de tipo estocástico y determinístico.

## 2.4 ¿POR QUÉ SIMULAR LOS PROCESOS?

- ❖ Simular permite diseñar correctamente los cambios complejos y conocer el efecto que estos cambios tienen en el sistema para poder solucionarlos de manera más efectiva.
- ❖ Permite eliminar inversiones innecesarias ya que se puede conocer las necesidades de equipo, transporte, almacenamiento y otros recursos con precisión.
- ❖ En Lean y Six Sigma ayuda a identificar y eliminar actividades que no generan valor agregado para el proceso, permite analizar el impacto de la variabilidad.
- ❖ Visualiza problemas y sus soluciones. Ayuda a comunicar ideas.
- ❖ Responde a las preguntas: “¿Qué pasa sí...?” antes de invertir capital y sin interrumpir la operación actual.

## 2.5 VENTAJAS Y DESVENTAJAS DE LA SIMULACIÓN

Según García, la simulación es una de las diversas herramientas con las que cuenta el analista para tomar decisiones y mejorar sus procesos. Sin embargo, es necesario destacar que, como todas las demás opciones de que dispone, la simulación de procesos presenta ventajas y desventajas que es preciso tomar en cuenta al determinar si es apta para resolver un problema determinado<sup>8</sup>.

---

<sup>8</sup> Ver: García, et al. 2006, p. 8.

Dentro de las ventajas más comunes que ofrece la simulación se encuentran las siguientes:

- ❖ Es muy buena herramienta para conocer el impacto de los cambios en los procesos sin necesidad de llevarlos a cabo en la realidad.
- ❖ Mejora el conocimiento del proceso actual al permitir que el analista vea cómo se comporta el modelo generado bajo diferentes escenarios.
- ❖ Puede utilizarse como medio de capacitación para la toma de decisiones.
- ❖ Es más económico realizar un estudio de simulación que hacer muchos cambios en los procesos reales.
- ❖ Permite probar varios escenarios en busca de las mejores condiciones de trabajo de los procesos que se simulan.
- ❖ Identificar oportunidades de mejora en el sistema bajo investigación.
- ❖ En problemas de gran complejidad, la simulación permite generar una buena solución.
- ❖ En la actualidad los paquetes de software para simulación tienden a ser más sencillos, lo que facilita su aplicación.
- ❖ Gracias a las herramientas de animación que forman parte de muchos de esos paquetes es posible ver cómo se comportará un proceso una vez que sea mejorado.

- ❖ Permite la experimentación en condiciones que podrían ser peligrosas o de elevado coste económico en el sistema real.
- ❖ Responde satisfactoriamente a preguntas hipotéticas del tipo “qué pasa si...”
- ❖ Es una técnica relativamente eficiente y muy flexible.
- ❖ En algunos casos la simulación es el único método disponible.

Entre las desventajas que puede presentar la simulación están:

- ❖ La simulación no es una herramienta de optimización. No genera soluciones óptimas a problemas de análisis cuantitativos.
- ❖ La simulación puede ser costosa cuando se quiere emplearla en problemas relativamente sencillos de resolver, en lugar de utilizar soluciones analíticas que se han desarrollado de manera específica para ese tipo de casos.
- ❖ Se requiere bastante tiempo, generalmente meses, para realizar un buen estudio de simulación.
- ❖ Es preciso que el analista domine el uso del paquete de simulación y que tenga sólidos conocimientos de estadística para interpretar los resultados.
- ❖ Existe el riesgo de tomar malas decisiones basadas en modelos de simulación que no han sido validados y verificados adecuadamente.

## 2.6 APLICACIONES DE LA SIMULACIÓN

De acuerdo con Guasch (2005), las aplicaciones más comunes de la simulación son:<sup>9</sup>

- ❖ **Procesos de fabricación.** Fue una de las primeras áreas beneficiadas por estas técnicas. La simulación se emplea tanto para el diseño como para la ayuda a la toma de decisiones operacionales.
- ❖ **Logística.** La simulación contribuye de forma significativa a la mejora de los procesos logísticos en general. Dentro de esta área, se incluye tanto una cadena completa de suministros, como la gestión de inventarios de un almacén.
- ❖ **Salud.** Se emplea tanto para la mejora de un departamento hospitalario, como en la logística asociada a los transplantes o a la coordinación médica de una región.
- ❖ **Negocios (Business Processing).** Simulación de los procesos administrativos y de negocio de una empresa. En esta área están teniendo mucho éxito los juegos de empresa en los que a través de la simulación de los efectos de las decisiones que se van tomando se puede entrenar a los directivos.
- ❖ **Servicios en general.** Servicios públicos, gestión de restaurantes, banca, empresas de seguros, etc.
- ❖ **Transporte y Distribución:** La simulación permite representar y analizar el comportamiento de sistemas de transporte tanto en redes metropolitanas (tráfico en las autopistas o en las ciudades, pertinencia de un semáforo en el

---

<sup>9</sup> Ver: Guasch, et al. 2005, p. 16.

cruce de dos vías) como en sistemas de fabricación (manejo y almacenamiento de producto en proceso).

- ❖ **Proyectos de Inversión:** la simulación es recomendada para el estudio de proyectos de inversión en los cuales la incertidumbre está asociada a la tasa de inflación, las tasas de interés, los flujos de efectivo etc., haciendo difícil y a veces imposible analizar analíticamente los flujos de caja.

## **2.7 PASOS PARA REALIZAR UN ESTUDIO DE SIMULACIÓN**

Como ocurre con otras herramientas de investigación, la realización de un estudio de simulación requiere la ejecución de una serie de actividades y análisis que permitan sacarle el mejor provecho. A continuación se mencionan los pasos básicos para realizar un estudio de simulación, según Blanco (2001) y Guasch, et al. (2005).

**2.7.1 Definición del sistema bajo estudio.** En esta etapa es necesario conocer el sistema a modelar. Para ello se requiere saber qué origina el estudio de simulación y establecer los supuestos del modelo: es conveniente definir con claridad las variables de decisión del modelo, determinar las interacciones entre éstas y establecer con precisión los alcances y limitaciones que aquel podría llegar a tener.

Antes de concluir este paso es recomendable contar con la información suficiente para lograr establecer un modelo conceptual del sistema bajo estudio, incluyendo sus fronteras y todos los elementos que lo componen, además de las interacciones entre éstos, flujos de productos, personas y recursos, así como las variables de mayor interés para el problema.

**2.7.2 Generación del modelo de simulación base.** Una vez que se ha definido el sistema en términos de un modelo conceptual, la siguiente etapa del estudio consiste en la generación de un modelo de simulación base. No es preciso que este modelo sea demasiado detallado, pues se requiere mucha más información estadística sobre el comportamiento de las variables de decisión del sistema.

**2.7.3 Recolección y análisis de datos.** De manera paralela a la generación del modelo base, es posible comenzar la recopilación de la información estadística de las variables aleatorias del modelo. En esta etapa se debe determinar qué información es útil para la determinación de las distribuciones de probabilidad asociadas a cada una de las variables aleatorias innecesarias para la simulación. Al finalizar la recolección y análisis de datos para todas las variables del modelo, se tendrán las condiciones necesarias para generar una versión preliminar del problema que se está simulando.

**2.7.4 Generación del modelo preliminar.** En esta etapa se integra la información obtenida a partir del análisis de los datos, los supuestos del modelo y todos los datos que se requieran para tener un modelo lo más cercano posible a la realidad del problema bajo estudio. Al finalizar esta etapa el modelo está listo para su primera prueba: su verificación o, en otras palabras, la comparación con la realidad.

**2.7.5 Verificación del modelo.** Una vez que se han identificado las distribuciones de probabilidad de las variables del modelo y se han implantado los supuestos acordados, es necesario realizar un proceso de verificación de datos para comprobar la propiedad de la programación del modelo, y comprobar que todos los parámetros usados en la simulación funcionen correctamente.

Una vez que se ha completado la verificación, el modelo está listo para su comparación con la realidad del problema que se está modelando. A esta etapa se le conoce también como validación del modelo.

**2.7.6 Validación del modelo.** El proceso de validación del modelo consiste en realizar una serie de pruebas al mismo, utilizando información de entrada real para observar su comportamiento y analizar sus resultados.

**2.7.7 Generación del modelo final.** Una vez que el modelo se ha validado, el analista está listo para realizar la simulación y estudiar el comportamiento del proceso. En caso de que se desee comparar escenarios diferentes para un mismo problema, éste será el modelo raíz; en tal situación, el siguiente paso es la definición de los escenarios a analizar.

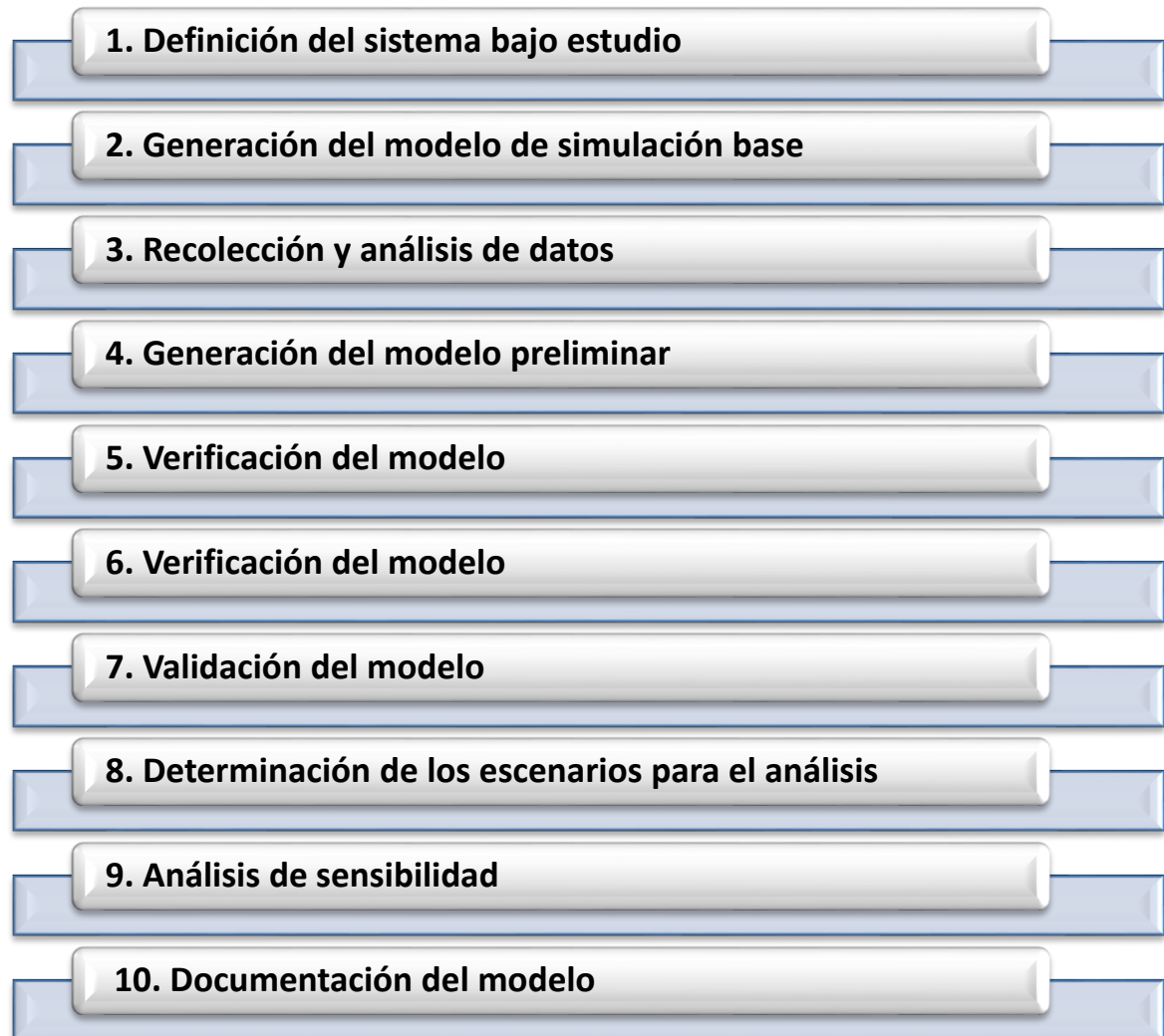
**2.7.8 Determinación de los escenarios para el análisis.** Tras validar el modelo es necesario acordar con el cliente los escenarios que se quiere analizar. Una manera muy sencilla de determinarlos consiste en utilizar un escenario pesimista, uno optimista y uno intermedio para la variable de respuesta más importante.

**2.7.9 Análisis de sensibilidad.** Una vez que se obtienen los resultados de los escenarios es importante realizar pruebas estadísticas que permitan comparar los escenarios con los mejores resultados finales.

**2.7.10 Documentación del modelo.** Sugerencias y conclusiones. Una vez realizado el análisis de los resultados, es necesario efectuar toda la documentación del modelo. Esta documentación es muy importante, pues permitirá el uso del modelo generado en caso de que se requieran ajustes futuros. En ella se deben incluir los supuestos del modelo, las distribuciones asociadas a sus variables, todos sus alcances y limitaciones y, en general, la totalidad de las consideraciones de programación.

En la Figura 4 se enuncian los pasos necesarios para realizar un estudio de simulación, según Blanco (2001) y Guasch, et al. (2005)

Figura 4. Pasos de una simulación



Fuente: Autor.

## 2.8 PROYECTOS DE SIMULACIÓN

Los proyectos de análisis y modelado de la simulación por lo general deben involucrar un equipo interdisciplinario de personas, aparte de los interesados en el proyecto, cada uno con unas funciones o roles específicos en cada una de las fases del proyecto vistas anteriormente. Estos roles por lo general son y tienen las siguientes actividades:

- ❖ **Desarrollador:** quien desarrolla y construye el modelo de simulación y da soporte al software.
- ❖ **Diseñador:** diseña el modelo y establece la arquitectura del sistema de soporte y decisión. Rol estrechamente vinculado con el desarrollador. Por lo general es quien diseña el modelo conceptual.
- ❖ **Analista:** define y diseña las variables de análisis de decisión de interés del sistema, la longitud de la corrida de simulación, número de réplicas, período de precalentamiento, escenarios a simular. Se necesita buen conocimiento estadístico para analizar y validar el modelo de simulación.
- ❖ **Investigador:** quien investiga y consulta si problemas similares han sido resueltos mediante simulación y otras técnicas cuantitativas y numéricas. Encargado de validar la precisión y fiabilidad de los datos disponibles.
- ❖ **Educador:** ayuda a capacitar a los interesados del proyecto para que se involucren activamente y puedan ser contribuidores efectivos. Es necesario porque normalmente los interesados no entienden de simulación, sus etapas, dificultades y por lo general son de diferentes profesiones y trasfondo cultural.

Aunque no es conveniente, estos roles pueden ser asumidos por una misma persona, si no todos la mayoría de ellos.

## **2.9 SIMULACIÓN MONTECARLO**

La simulación Monte Carlo (MC) es una técnica fundamentada en conceptos estadísticos como muestreo aleatorio y la capacidad que tienen los computadores para generar números pseudo-aleatorios y automatizar cálculos que permiten resolver problemas matemáticos mediante la simulación de variables aleatorias.

John Von Neumann, a finales de los años 40 y con los primeros computadores, aplica la simulación MC para resolver problemas complejos que no podían ser resueltos de forma analítica.

La idea es crear un modelo matemático del sistema, proceso o actividad analizada, ubicando variables identificadas como inputs o entradas del modelo para:

- ❖ Generar muestras aleatorias o valores concretos.
  
- ❖ Analizar el comportamiento del sistema ante los valores generados.

Luego de repetir  $n$  veces este experimento, se dispone de  $n$  observaciones del comportamiento del sistema, para realizar un análisis detallado. Entre mayor sea el número  $n$  de experimentos, más preciso será el estudio.

Esta es una técnica ampliamente utilizada como alternativa a modelos matemáticos exactos y para estimar soluciones de problemas complejos.

## 2.10 GENERACIÓN DE NÚMEROS ALEATORIOS

Según Ríos (2005), la habilidad para reproducir el comportamiento aleatorio o variación propia de un sistema estocástico es una de las poderosas características que posee la simulación. Es por esto que para el desarrollo simulado de un sistema aleatorio se requiere contar con un método generador de números aleatorios, así como también modelos generadores de variables aleatorias basadas en distribuciones de probabilidad conocidas.<sup>10</sup>

Según Ross (1999), el núcleo de un estudio de Simulación es la capacidad de generar números aleatorios que representan el valor de una variable aleatoria.

Esto se realiza en dos pasos. El primero consiste en generar una secuencia de números aleatorios con distribución uniforme entre 0 y 1. El segundo consiste en transformar esta secuencia para obtener los valores aleatorios según las distribuciones deseadas.

Dado que no es posible generar números de naturaleza aleatoria, se consideran números pseudo aleatorios, generados por medio de algoritmos determinísticos que requieren un parámetro de inicio.

Por lo tanto para garantizar altos niveles de confiabilidad en el comportamiento aleatorio del conjunto de números que se utiliza en una simulación es necesario asegurar que el generador de dichos números aleatorios cree un conjunto de números con propiedades de independencia y uniformidad.

- ✓ Propiedad de Independencia: Implica que los números aleatorios no deben tener correlación alguna entre sí, donde la probabilidad de observar un

---

<sup>10</sup> Ver: Ríos, et al. 2009, cap. 2.

valor en un intervalo particular es independiente de los valores previamente generados.

- ✓ Propiedad de Uniformidad: Si el intervalo  $(0,1)$  es dividido en  $n$  clases o sub intervalos de igual longitud, el numero esperado de observaciones en cada intervalo es  $N/m$ , donde  $N$  es el número total de observaciones.

Para validar si los números que integran un conjunto determinado son de tipo aleatorio y pueden usarse en un estudio de simulación, es posible realizar una serie de pruebas estadísticas que corroboren la independencia y uniformidad de los datos estudiados, además de garantizar su distribución uniforme.

## **2.11 GENERACIÓN DE VARIABLES ALEATORIAS**

Para la mayor comprensión conceptual en problemas de simulación resulta importante reconocer la utilidad de algunos métodos para la generación de variables aleatorias. Existe un gran número de referencias que abordan el tema de la generación de variables aleatorias de manera numérica. El propósito de este tema es identificar algunas técnicas generales que permitan generar una variable aleatoria a partir de algunas distribuciones de probabilidad comunes.

Existen varios métodos que permiten generar variables aleatorias. Lo normal es que existan varias opciones para generar una misma variable aleatoria. La elección del método adecuado se puede basar en una serie de factores como:

- ❖ Exactitud: se han de obtener valores de una variable con una precisión dada. A veces se tiene suficiente con obtener una aproximación y otras no. Se prefiere un método exacto frente a métodos aproximados, como soluciones numéricas.

- ❖ Eficiencia: el algoritmo que implementa el método de generación tiene asociado un tiempo de ejecución y un gasto de memoria. Elegiremos un método que sea eficiente en cuanto al tiempo y a la cantidad de memoria requeridos.
- ❖ Espacio. Necesidades de memoria del método utilizado. En general, los métodos no consumen mucha memoria.
- ❖ Complejidad: Buscamos métodos que tengan complejidad mínima, siempre y cuando se garantice cierta exactitud.
- ❖ Robustez: el método tiene que ser eficiente para cualquier valor que tomen los parámetros de la distribución que siga la variable aleatoria.
- ❖ Facilidad de implementación. Simplicidad.

La mayoría de las técnicas utilizadas para la generación de variables aleatorias se pueden agrupar en:

- ❖ Método de la transformada inversa.
- ❖ Método de aceptación-rechazo.
- ❖ Método de composición.
- ❖ Método de convolución.

A continuación se presenta un ejemplo de cada uno de los métodos.

### 2.11.1 Método de la Transformada Inversa

Es el método más directo para generar una variable aleatoria y el más utilizado en experimentos de simulación. El método de la transformación inversa se puede utilizar para generar variables aleatorias a partir de una distribución continua. Se aprovecha el hecho de que la función de distribución acumulada es uniforme en el intervalo (0, 1).

$$U = F(X).$$

Si  $U$  es una variable aleatoria uniforme en (0, 1), entonces se puede obtener la variable aleatoria  $X$  a partir de la siguiente relación

$$X = F^{-1}(U).$$

El procedimiento general para el método de la transformación inversa es el siguiente

1. Obtener una expresión para la función de distribución inversa  $F^{-1}(U)$ .
2. Generar un número uniforme  $U$ .
3. Entregar la variable aleatoria  $X$  deseada, usando  $X = F^{-1}(U)$ .

#### Algoritmo del método de la transformada inversa:

**Propósito:** Generar  $Z$  aleatoriamente de  $F(z)$ ,  $a \leq z \leq b$ .

**Entrada:** Capacidad para evaluar  $F^{-1}(u)$ ,  $0 \leq u \leq 1$ .

**Salida:**  $Z$

**Método:** Generar aleatoriamente  $U$  de  $u(0, 1)$

$Z \leftarrow F^{-1}(U)$

Devolver  $Z$ .

### Ejemplo 1: La distribución exponencial.

La función de distribución acumulada para una variable aleatoria exponencial con parámetro  $\lambda$  está dada por

$$F(x) = 1 - e^{-\lambda x}; 0 < x < 1.$$

Haciendo

$$u = F(x) = 1 - e^{-\lambda x},$$

Se puede resolver para encontrar  $x$  como sigue

$$\begin{aligned} u &= 1 - e^{-\lambda x} \\ e^{-\lambda x} &= 1 - u \\ -\lambda x &= \ln(1 - u) \\ X &= -\frac{1}{\lambda} \ln(1 - u) \end{aligned}$$

Notando que la cantidad  $1 - u$  también se distribuye uniformemente en el intervalo  $(0, 1)$ , se puede generar una variable aleatoria exponencial con parámetro  $\lambda$  usando la transformación

$$X = -\frac{1}{\lambda} \ln(1 - u)$$

La misma técnica de la transformación inversa se puede emplear para el caso discreto. Por ejemplo, si se desea generar una variable aleatoria discreta  $X$  que tenga una función de densidad de probabilidad dada por:

$$P(X = x_i) = p_i; x_0 < x_1 < x_2 < \dots; \sum_{i=0}^N P_i = 1.$$

La variable aleatoria se obtiene generando un número aleatorio  $U$  distribuido uniformemente, y después se calcula la variable  $X$  de acuerdo al siguiente criterio

$$X = x_i \quad \text{si } F(x_{i-1}) < U \leq F(x_i)$$

### **Algoritmo del método de la transformada inversa para una variable discreta**

El algoritmo de este procedimiento es el siguiente:

1. Definir una función de probabilidad para  $x_i$ ,  $i = 1, \dots, k$ . Tener en cuenta que  $k$  puede crecer indefinidamente.
2. Generar un número aleatorio uniforme  $U$ .
3. Si  $U \leq p_0$  entrega  $X = x_0$
4. de otro modo si  $U \leq p_0 + p_1$  entrega  $X = x_1$
5. de otro modo si  $U \leq p_0 + p_1 + p_2$  entrega  $X = x_2$
6. de otro modo si  $U \leq p_0 + \dots + p_k$  entrega  $X = x_k$ .

### **2.11.2 Método de Aceptación y Rechazo**

En algunos casos se puede tener un método más simple para generar una variable aleatoria a partir de una densidad de probabilidad, por ejemplo  $g(y)$ , en lugar de la densidad que se está buscando. Se puede utilizar esta densidad para generar la variable a partir de la densidad continua  $f(x)$  que se desea. Primero se genera un número aleatorio  $Y$  a partir de  $g(y)$  y se acepta el valor con una probabilidad que es proporcional al ratio  $f(Y)/g(Y)$ .

Si se define  $c$  como una constante que satisface

$$\frac{f(y)}{g(y)} \leq c \text{ para toda } y \in \mathbb{R} \text{ tal que } f(y) \neq 0 \quad (1)$$

Entonces se puede generar la variable deseada usando el siguiente procedimiento

1. Escoger una densidad  $g(y)$  partir de la cual es fácil obtener una muestra.
2. Encontrar una constante  $c$  tal que la ecuación (1) se satisface.
3. Generar un número aleatorio  $Y$  a partir de la densidad  $g(y)$ .
4. Generar un número aleatorio uniforme  $U$ .
5. Si se cumple que  $U \leq \frac{f(y)}{c g(y)}$

Entonces aceptar  $X = Y$ , de otro modo ir al paso 3.

### **Algoritmo: Método de aceptación y rechazo**

#### ***Repetir***

*Generar  $Y$ , con densidad  $g$ ;*

*Generar  $U \sim \mathcal{U}(0; 1)$*

***Hasta***  $U < f(Y) / (cg(Y))$  ;

$X \leftarrow Y$

### 2.11.3 Método de Convolución

Muchas variables aleatorias incluyendo la normal, binomial, poisson, gamma, erlang, etc., se pueden expresar de forma exacta o aproximada mediante la suma lineal de otras variables aleatorias.

El método de convolución se puede usar siempre y cuando la variable aleatoria  $x$  se pueda expresar como una combinación lineal de  $k$  variables aleatorias:

$$x = b_1x_1 + b_2x_2 + \dots + b_kx_k$$

En este método se necesita generar  $k$  números aleatorios  $(u_1, u_2, \dots, u_k)$  para generar  $(x_1, x_2, \dots, x_k)$  variables aleatorias usando alguno de los métodos anteriores y así poder obtener un valor de la variable que se desea obtener por convolución.

Si dos variables aleatorias ( $X$  e  $Y$ ) son independientes, la función densidad de probabilidad de la suma  $Z=X+Y$  puede demostrarse que es:

$$f_z(z) = f_x(z) * f_y(z)$$

De ahí el nombre del método.

Si se sabe que  $Z$  cumple esta propiedad, por qué no:

- Generar muestras (en el ejemplo  $x_i$  e  $y_i$ ) de las variables aleatorias originales ( $X$  e  $Y$  en el ejemplo) utilizando el método que sea preciso.
- Sumarlas (en el ejemplo,  $z_i=x_i+y_i$ )

## 2.12 TÉCNICAS DE REDUCCIÓN DE VARIANZA

A medida que aumenta la complejidad en la simulación de un sistema, aumenta también de manera considerable el tiempo de procesamiento de la simulación, por tanto, es necesario emplear técnicas que permitan conseguir que el intervalo de confianza de las estimaciones sea mejor para un mismo tamaño de muestra  $n$ , o, de forma equivalente, reducir el tamaño muestral necesario para alcanzar un intervalo de confianza especificado.

En las dos situaciones se mejora la eficiencia de la simulación, ya que se consigue una mejoría de resultados en un tiempo de ejecución no superior al de partida, garantizando una relación inversa entre el tiempo de ejecución y el error cometido en las estimaciones. Para lograr cualquiera de los dos objetivos comentados se deben conseguir estimadores de menor varianza de los estadísticos de los procesos estocásticos bajo estudio.<sup>11</sup>

Las técnicas de reducción de varianza intentan reducir la incertidumbre en los datos de salida, aunque por lo general es imposible conocer con antelación si el método tendrá éxito o no. La aplicación de una u otra técnica depende de las características del modelo.

Tres de las técnicas más conocidas son:

1. Variables antitéticas.
2. El método de la variables de control
3. Números aleatorios comunes.

---

<sup>11</sup> Ver: Guasch, et al. 2005, cap. 3.

### 2.12.1 Variables antitéticas

Este método consiste en ejecutar dos simulaciones en paralelo con secuencias de números aleatorios complementarios, con el fin de que ambas simulaciones estén correlacionadas negativamente. Como resultado se toma el promedio de las dos observaciones porque está más próximo al valor esperado  $\mu$  que cada de una de las observaciones por separado.

El método introduce la correlación negativa mediante el uso de números aleatorios complementarios para la generación de muestras de las variables de entrada a las dos simulaciones. Es decir, si la primera simulación usa una variable aleatoria uniformemente distribuida entre 0 y 1,  $U \sim U(0, 1)$ , para la generación de muestras utilizando el método de la transformada inversa, se utilizará la variable  $U' = 1 - U$  en la segunda simulación y que también está uniformemente distribuida entre 0 y 1. Por ejemplo si se utiliza un valor  $u_i$  para generar una muestra del tiempo de proceso de una máquina en la simulación 1, entonces se utilizará el valor  $1 - u_i$  para el tiempo de proceso de esa máquina en la simulación 2.

Para el funcionamiento del método es vital la sincronización entre las dos simulaciones.

### 2.12.2 Variables de control

Este método consiste en aprovechar la correlación entre varias variables aleatorias para obtener una reducción de la varianza del estimador de la media en una de ellas.

Se estima la media  $\mu$  de una variable aleatoria  $X$ ; se conoce el valor esperado de otra variable aleatoria  $Y$  de la simulación,  $v = E[Y]$  y por último se sabe que  $X$  e  $Y$  están correlacionadas, positiva o negativamente.

Si la correlación es positiva, entonces valores de  $Y > v$  tenderán a ir acompañados de valores de  $X > \mu$  y viceversa. La aplicación del método consiste en observar valores en la simulación de  $Y > v$ , entonces se puede sospechar que  $X > \mu$  y con esta información es posible corregir el valor de  $X$ , disminuyéndolo en cierta cantidad. Si se produce el efecto contrario se aumenta el valor de  $X$ .

Se utiliza el conocimiento que se tiene sobre la desviación de  $Y$  respecto de su media para acercar  $X$  hacia la suya  $\mu$ , reduciendo así su variabilidad. Se dice, entonces, que  $Y$  es la variable de control de  $X$ .

Para definir la cantidad necesaria para ajustar el valor de  $X$ , se define un estimador controlado  $X_c$  como:

$$X_c = X - a * (Y - v),$$

Donde  $a$  es un valor constante que tiene el mismo signo de la correlación entre las dos variables aleatorias  $X$  e  $Y$ .

Una buena variable de control debe estar altamente correlacionada con  $X$ , para que ofrezca la mayor información posible sobre las variaciones de  $X$ , además de presentar una varianza reducida. Para saber que variable de control debe utilizarse en la simulación se debe analizar la estructura del sistema o recurrir a la experimentación sobre el modelo.

### **2.12.3 Números aleatorios comunes**

Este método consiste en comparar dos o más configuraciones alternativas para un mismo sistema, en lugar de analizar el comportamiento con una única configuración. Se deben comparar bajo condiciones experimentales similares. En un modelo de simulación las condiciones experimentales son generadas por las variables aleatorias que modelan las circunstancias ambientales bajo las que se encuentra el sistema.

Si se usa una muestra de la variable aleatoria  $U \sim U(0, 1)$  en una de las simulaciones, deberá utilizarse la misma muestra para el mismo propósito en la otra simulación o alternativa, consiguiendo introducir correlación positiva utilizando la misma secuencia de números aleatorios en ambas simulaciones.

Al igual que con la técnica de variación antitética, es vital mantener la sincronización para que la técnica funcione y se logre la reducción de varianza deseada.

### 3. SOFTWARE DE SIMULACIÓN

Un modelo de simulación puede ser programado en cualquier lenguaje de programación conocido, como C++, Matlab, Java, sin embargo existen aplicaciones especializadas en la programación de ciertos tipos de modelos, conocidas como simuladores o software de simulación.

Entre las ventajas de un software de simulación, se encuentra que contienen elementos predefinidos que facilitan el modelado del sistema de estudio y están organizados en módulos que permiten realizar tareas comunes para la simulación de cualquier modelo.

Por lo tanto la simulación de un sistema puede programarse utilizando:

- ❖ Lenguajes de propósito general.
  
- ❖ Lenguajes de Simulación, que son de propósito específico.

Las diferencias radican en la flexibilidad y facilidad de uso que proporcionan cada uno. Los lenguajes de programación son más complejos y tienen pocas funciones específicas para simulaciones, por lo que el modelador necesitará crear sus propias rutinas para la simulación implicando un mayor esfuerzo en la construcción del modelo. En cambio un software de simulación está diseñado para este propósito y facilitar su utilización, dado que la mayoría de estas aplicaciones tienen funciones predefinidas para simular sistemas productivos, logísticos, de servicios, además de contar con una interfaz gráfica de usuario donde se puede utilizar y visualizar elementos gráficos para la animación y una mejor presentación de los resultados de la simulación.

### 3.1 COMPONENTES DE UN SOFTWARE DE SIMULACIÓN

- ❖ **Módulo de ajuste de datos de entrada:** permite realizar el ajuste de los datos a una distribución de probabilidad teórica conocida mediante pruebas de bondad de ajuste. Permite importar, exportar datos a otras aplicaciones, por ejemplo Excel. Todo software de simulación actualizado tiene este módulo.
  
- ❖ **Módulo de interface de la modelación:** permite al usuario caracterizar el modelo mediante las herramientas gráficas, de diálogo y texto, así como archivos externos de datos que definan el modelo.
  
- ❖ **Procesador del modelo:** Crea una base de datos del modelo que contiene datos de entrada. Si es necesario realiza conversiones de los datos que permitan iniciar la simulación de manera eficiente.
  
- ❖ **Módulo de interface de la simulación:** despliega la animación que se visualiza durante la ejecución de la simulación y que permite al usuario interactuar con la simulación para controlar efectos de la animación como su velocidad, ver el estado de las variables, entre otros.
  
- ❖ **Procesador de la simulación:** se encarga de ejecutar la lógica de los eventos y actualizar las estadísticas que se van obteniendo mientras se ejecuta la simulación. Se compone de elementos como :
  - ✓ Variable reloj: lleva el control del avance del tiempo.
  
  - ✓ Calendario de eventos: consiste en una lista que contiene los eventos que se ejecutarán en el orden cronológico en que ocurren.

- ✓ Lógica del evento: contiene las instrucciones para simular el comportamiento de un evento.
  - ✓ Generador de números aleatorios: algoritmos para calcular valores para variables aleatorias.
- 
- ❖ **Procesador de animación:** interactúa con la base de datos de la simulación para actualizar la representación gráfica de lo que se está simulando. La animación se despliega durante la ejecución de la simulación dando una visión cualitativa de lo que sucede en el modelo pero los resultados importantes de la simulación consisten en resúmenes estadísticos, por lo que la animación es un complemento de la presentación de resultados.
  - ❖ **Procesador de salida:** resume los datos estadísticos calculados durante la simulación y crea bases de datos conteniendo estos resultados.
  - ❖ **Interface de salida:** proporciona la interface para que el usuario pueda visualizar de manera clara los resultados de la simulación.
  - ❖ **Módulo de Optimización:** tiene como objetivo mejorar el diseño del modelo a través de una gestión eficiente de los recursos y actividades, partiendo de una función objetivo ( $Z$ ) definida por el usuario, y las restricciones (sujeto a) automáticamente crea los escenarios posibles de la simulación y los ejecuta, obteniendo el resultado óptimo. Para un usuario del modelo, el mejor modelo podría ser: maximizar unidades producidas, minimizando costo; tal vez para otro usuario la mejor función objetivo es: maximizar unidades producidas, minimizando inventario en proceso y maximizando la utilización de los operarios.

### 3.2 SOFTWARE EMPLEADO EN LA ACADEMIA E INDUSTRIA

Según la Winter Simulation Conference (WSC)<sup>12</sup>, un foro único a nivel internacional que reúne a los más reconocidos intelectuales que trabajan alrededor de la Simulación Dinámica Computacional para que publiquen los logros obtenidos a través de esta herramienta tecnológica, Arena es el software de simulación más utilizado tanto en la industria como en la academia.

En Colombia más del 50% de los programas de Ingeniería Industrial están usando Flexsim, según información suministrada por Vatic Consulting Group<sup>13</sup> en su página web, es decir, más de 83 programas de Ingeniería Industrial a nivel de pregrado de los 166 que hay en Colombia<sup>14</sup>.

Estos dos programas de simulación junto con ProModel son los más utilizados en la simulación de eventos discretos, gracias a su flexibilidad y facilidad en la construcción de modelos, aunque difiriendo considerablemente en el paradigma de construcción de modelos de sistemas, sobre todo Arena respecto de ProModel y Flexsim, dado que estos últimos se apoyan en un motor gráfico y un aspecto visual que facilite la construcción de simulaciones.

Con respecto a la literatura disponible para el aprendizaje de estos programas, Arena y ProModel son los que gozan de mayor literatura.

Entre los principales software de simulación de eventos discretos se encuentran:

---

<sup>12</sup> WSC: <http://www.wintersim.org/>

<sup>13</sup> Empresa Colombiana especializada en consultoría logística e investigación de operaciones y distribuidora del software Flexsim para la región andina.

<sup>14</sup> Consultado en: Módulo de consulta del SNIES - Sistema Nacional de Información de Educación Superior del MEN. Recuperado el 2 de julio de 2012 en <http://www.mineducacion.gov.co/sistemasdeinformacion/1735/w3-article-218261.html>

### 3.2.1 FLEXSIM

Tabla 1. Software Flexsim.

	
<b>Software</b>	Flexsim v. 6.0.2
<b>Empresa</b>	Flexsim Software Products, Inc
<b>Clasificación</b>	Eventos Discretos y Continuos
<b>Módulos</b>	ExpertFit, OptQuest

Fuente: FlexSim Software Products, Inc.

Es un software de simulación de eventos discretos y continuos basado en el paradigma de construcción y programación orientada a objetos y modelado basado en la animación (3D). La última versión disponible a Enero de 2013 es la 6.0.2. Utiliza el software ExpertFit para el ajuste y análisis de datos a distribuciones de probabilidad y la aplicación OptQuest para optimizar los modelos de simulación.

Es desarrollado por FlexSim Software Products, Inc., fundada en 1993 por Bill Nordgren, cofundador de ProModel Corporation y opera en Orem, Utah, Estados Unidos. Este simulador nace en el 2003 producto de una evolución en el paquete de simulación Taylor II y Taylor ED.

Flexsim presenta un impactante ambiente tridimensional y cuenta con una flexibilidad de simular cualquier tipo de proceso y construir modelos de simulación con la mayor rapidez y con una facilidad de uso extraordinaria, sin necesidad de programación, aunque recibe código C/C++ y tiene un lenguaje propio llamado FlexScript, lo cual lo convierte en uno de los software de simulación o el más

robusto en el mercado. Permite evaluar diversos escenarios y responder a preguntas del tipo ¿qué pasa si...? de una manera rápida, precisa y lo más importante: libre de riesgos. Además la visualización de una simulación tridimensional animada es la manera más efectiva de explicar un nuevo concepto o proyecto. Un modelo de simulación en tres dimensiones impresiona y capta la atención de los asistentes, permite exponer tanto internamente como externamente un concepto o sistema propuesto y marcar la diferencia con la competencia.

La visualización en 3D no es como en los demás software de simulación que utiliza una vista 2D de una perspectiva en 3D que solo permite alejarse y acercarse dando la ilusión que es en 3D cuando no lo es, sino que realmente permite ver todo o cualquier parte del proceso desde cualquier ángulo, rotar y hacer acercamientos.

Este simulador permite visualizar y probar cambios en las operaciones en cualquier proceso de manufactura, logística, manejo de materiales y de servicios, evitando los altos costos riesgos y extensos tiempos que conllevan el experimentar con cambios en el mundo real y su análisis por prueba y error, generando grandes ahorros. Cuenta con gráficos, reportes y estadísticas como resultados del modelo de simulación, además de importar datos de entrada desde cualquier programa o base de datos. También cuenta con la capacidad de crear videos de las simulaciones desde el mismo Flexsim que posteriormente se pueden presentar o compartir. Los objetos y el layout utilizados en el modelo pueden ser importados desde aplicaciones CAD, o utilizar la librería 3D que trae por defecto, tanto para modelos discretos como sistemas continuos.

La empresa Flexsim cuenta también con los siguientes productos:

- ❖ Flexsim CT o Flexsim Container Terminal: para simulación de puertos.
- ❖ Flexsim HC o Flexsim HealthCare: para simulación en el área de salud.
- ❖ Flexsim DS o Flexsim Distributed Simulation: para simulación distribuida, principalmente para fines militares.

Entre sus principales clientes en la industria se encuentran:

Coca-Cola, Whirlpool, NASA, General Mills, DuPont, Goodyear, Puerto de Cartagena, Siemens, Maersk, Norfolk Southern, Pfizer, Avon, Volkswagen, Ford, Mattel, Michelin, FedEx, Gillette y muchos otros.

## **COMPLEMENTOS**

### ❖ **ExpertFit**

Es un módulo para realizar pruebas de bondad de ajuste, el cual determina cual es la distribución de probabilidad que mejor se ajusta, como podría ser una distribución normal indicando los valores de la media y su desviación estándar o bien una exponencial por mencionar un ejemplo.

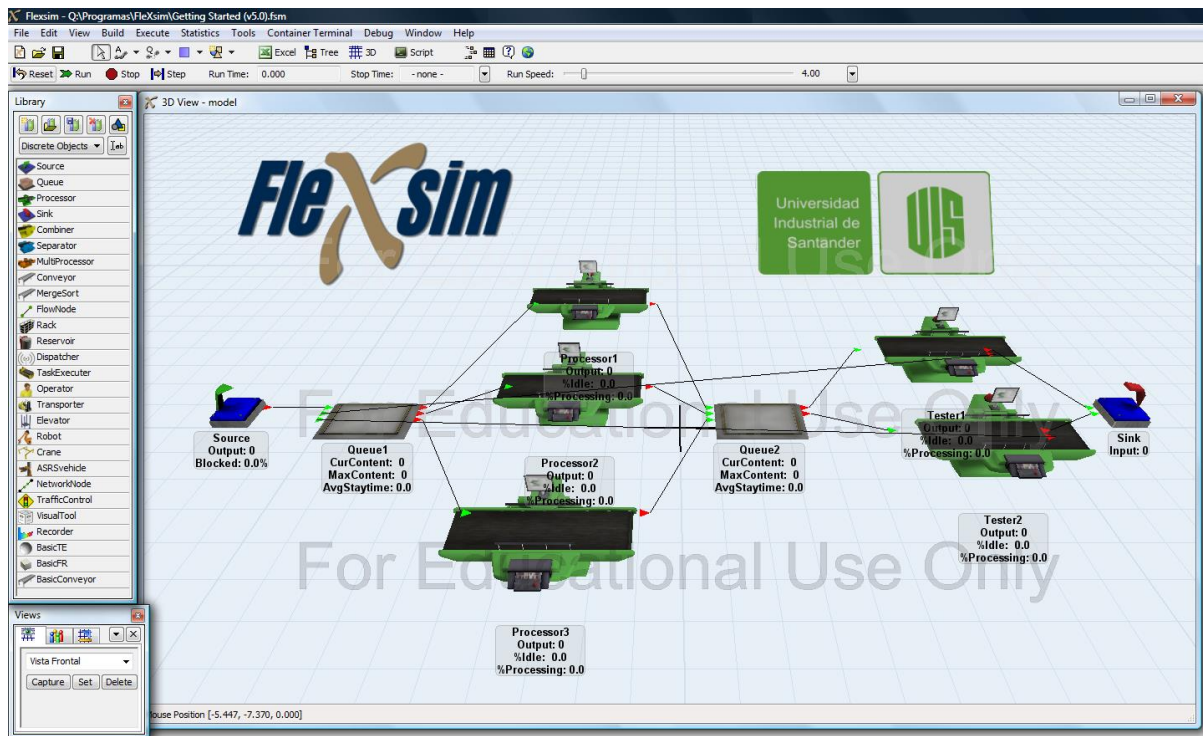
### ❖ **OptQuest**

El módulo de optimización y diseño de experimentos para los modelos de simulación automáticamente crea diferentes escenarios a partir de los modelos creados en Flexsim variando las distintas condiciones que se le especifiquen tales como el número de personas, montacargas, inventario, capacidad de las máquinas, velocidades de las bandas transportadoras, personal de mantenimiento, turnos, demandas o cualquier variable que se desee, utilizando algoritmos para evaluar esta gama de escenarios y encontrar automáticamente

la solución óptima. Los criterios para encontrar esta solución óptima los define el usuario y pueden ser cualquiera de los objetivos que se deseen.

Ejemplos de los objetivos son maximizar utilidades, minimizar tiempos de surtido, minimizar los tiempos de espera, maximizar piezas producidas, maximizar el nivel de servicio o cualquier otro objetivo que se desee.


Figura 5. Ejemplo de Modelo de Simulación en Flexsim.



Fuente: Introducción a Flexsim. Guía del Usuario de Flexsim.

### 3.2.2 PROMODEL

Tabla 2. Software ProModel.

	
<b>Software</b>	ProModel
<b>Empresa</b>	ProModel Corporation
<b>Clasificación</b>	Eventos Discretos
<b>Módulos</b>	StatFit, SimRunner (v7.5 y anteriores), OptQuest versiones recientes

Fuente: ProModel Corporation.

Es un software de simulación de eventos discretos que también permite modelar procesos continuos, basado en el paradigma de construcción y programación orientada a objetos y modelado basado en la animación. Desarrollado por ProModel Corporation, opera en Orem, Utah, Estados Unidos.

Puede simular Justo a Tiempo (JIT), Teoría de Restricciones (TOC), Sistemas de Empujar (Push), Halar (Pull), Logística, etc. Prácticamente, cualquier sistema puede ser modelado.

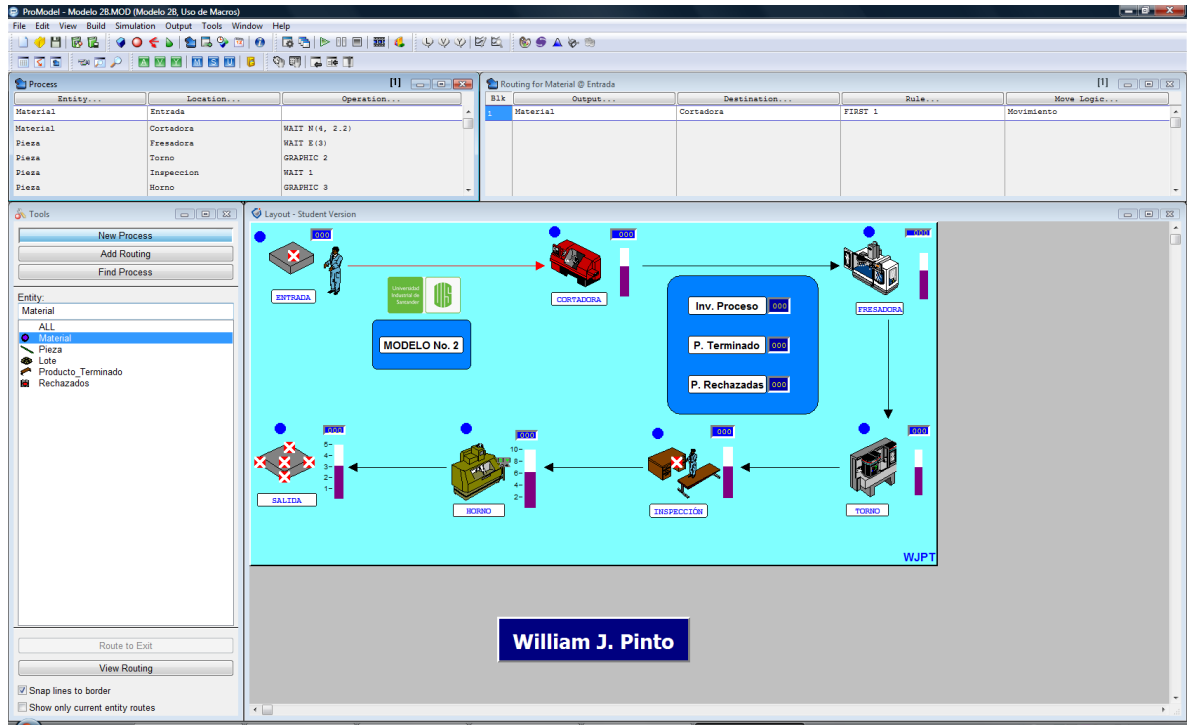
Incluye el optimizador SimRunner en las versiones anteriores a 2008 y OptQuest en las versiones recientes. Las pruebas de bondad de ajuste son realizadas con el módulo StatFit.

ProModel Corporation ofrece también los siguientes productos:

- ❖ ServiceModel: para simular industrias de servicios.
- ❖ MedModel: específicamente diseñada para el área de salud.
- ❖ Process Simulator: que agrega capacidades de simulación y análisis a Microsoft Visio.
- ❖ Project Simulator: es un plug-in de escenario de planificación, simulación y presentación de informes para Microsoft Project que permite a los administradores de proyectos visualizar rápidamente, analizar y optimizar sus planes de proyecto, resultando en una mayor probabilidad de completar proyectos a tiempo y dentro del presupuesto.

Sus principales clientes son: Pfizer, Lockheed Martin, U.S Army, TSA, Merck, West Point, General Dynamics, Medtronic, Baxter BioPharma, Driscoll's, Federal Express, Northrop Grumman, Boeing, MIT, U.S. Air Force, Nestle Purina.

Figura 6. Ejemplo de Modelo de Simulación en ProModel.



Fuente: GARAVITO, E.; ARENA, P. Simulación de Procesos de Manufactura 1. Publicaciones UIS, Bucaramanga, 2008.


Figura 7. Gráficos estadísticos en ProModel.



Fuente: ProModel Corporation.

### 3.2.3 ARENA

Tabla 3. Arena.

	
<b>Software</b>	Arena v. 14
<b>Empresa</b>	Rockwell Automation
<b>Clasificación</b>	Eventos Discretos y Continuos
<b>Módulos</b>	Input Analyzer, Output Analyzer, OptQuest

Fuente: Rockwell Automation.

Es un Software de simulación y automatización desarrollado por Rockwell Automation. Utiliza un paradigma de modelado basado en diagrama de flujo. La última versión disponible a Enero de 2013 es la 14 que incluye animación en 3D.

Arena utiliza el lenguaje de simulación y procesador SIMAN. Actualmente es el software de simulación más utilizado tanto en la industria como en la academia.

El Software Arena® es la herramienta de simulación de eventos discretos y continuos líder a nivel mundial<sup>15</sup>.

Las empresas más exitosa usan Arena® dado que es un software de simulación no solamente orientado a procesos productivos, sino también a servicios en un solo programa, sin depender de módulos adicionales.

---

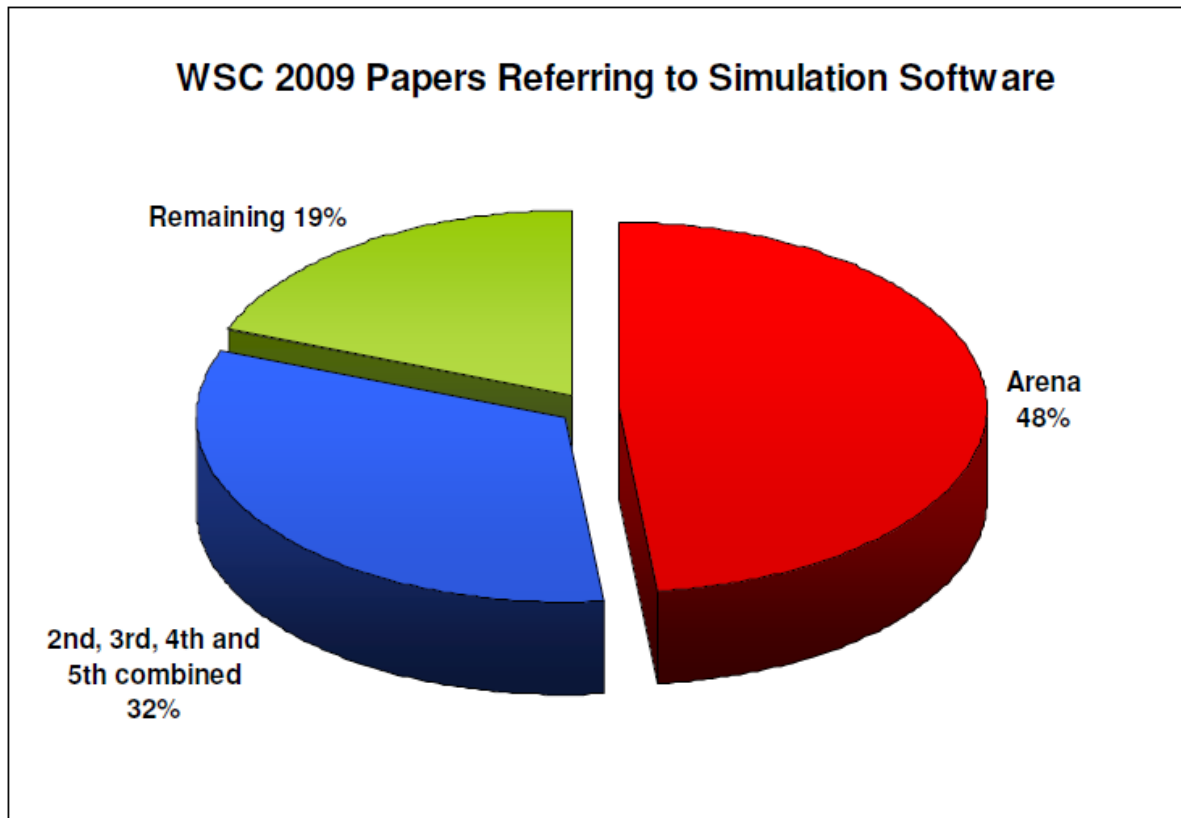
<sup>15</sup> Tomado de: <http://www.mimesis-soluciones.com/gpage.html> Recuperado el 10 de agosto de 2011.

Entre los usuarios académicos nacionales encontramos la Universidad de Pamplona, Universidad Central, Universidad Nacional, Universidad de los Andes, Universidad Industrial de Santander, Politécnico Gran Colombiano, Universidad EAFIT, Universidad del Magdalena. Universidades en USA que usan Arena son:

Massachusetts Institute of  
Technology  
University of Wisconsin  
Harvard  
Rensselaer Polytechnic Institute  
Columbia University  
University of Washington  
University of Pennsylvania  
The Pennsylvania State University  
Dartmouth College  
University of Florida  
Northwestern University  
University of Texas - Austin  
Cornell University  
University of Miami  
Carnegie Mellon University

George Washington University  
University of Virginia  
Ohio State University  
University of Southern California  
Boston University  
University of Michigan  
University of Pittsburgh  
University of North Carolina  
University of Georgia  
College of William & Mary  
Clemson University  
Lehigh University  
Purdue University  
University of Illinois – Urbana-  
Champaign  
Texas A&M University

Figura 8. Documentos de WCS<sup>16</sup> haciendo referencia a Arena.



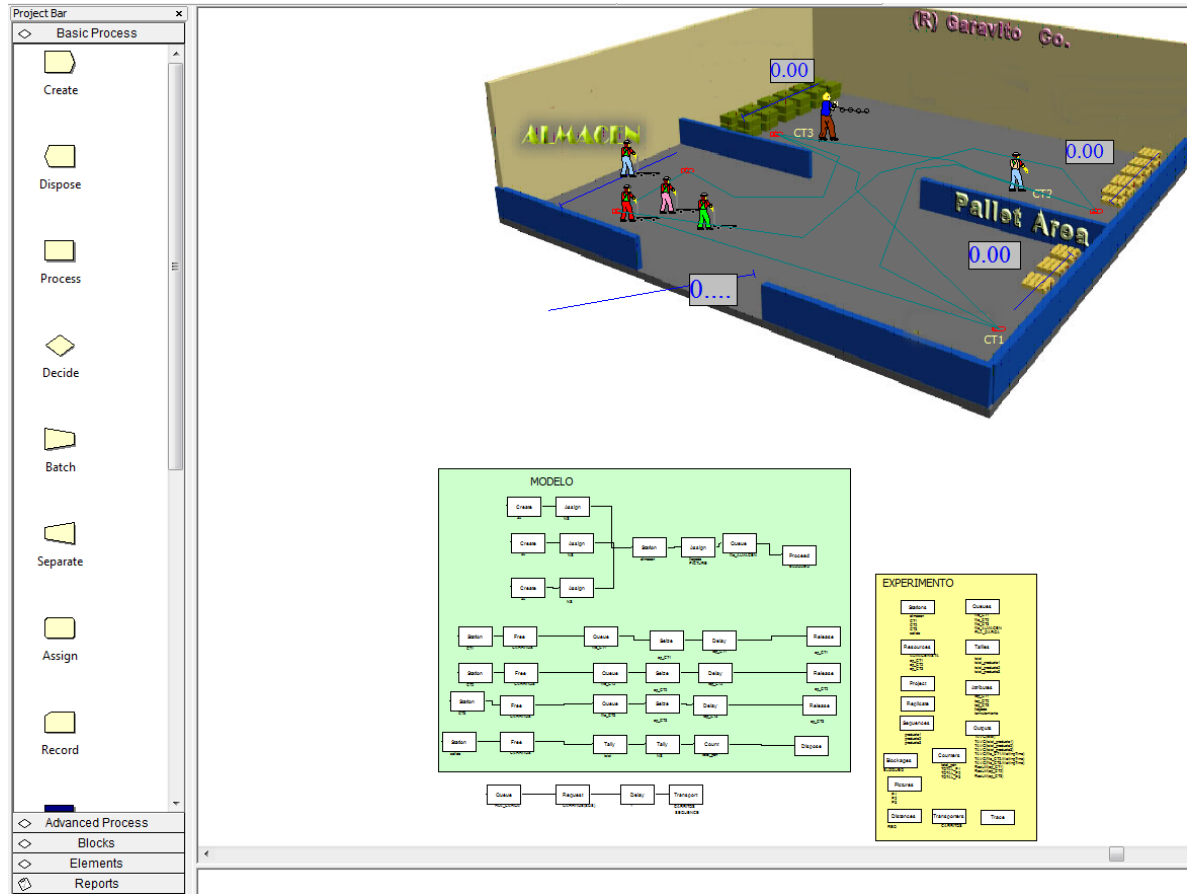
Fuente: Rockwell Automation.

Arena tiene un módulo de optimización *OptQuest* al igual que otros lenguajes de simulación y un módulo de entrada de datos para realizar pruebas de bondad de ajuste *Input Analyzer*.

---

<sup>16</sup> WCS: Winter Simulation Conference.

Figura 9. Modelo de Simulación en Arena.





Fuente: Modelo de clase TMO.

### 3.2.4 WITNESS

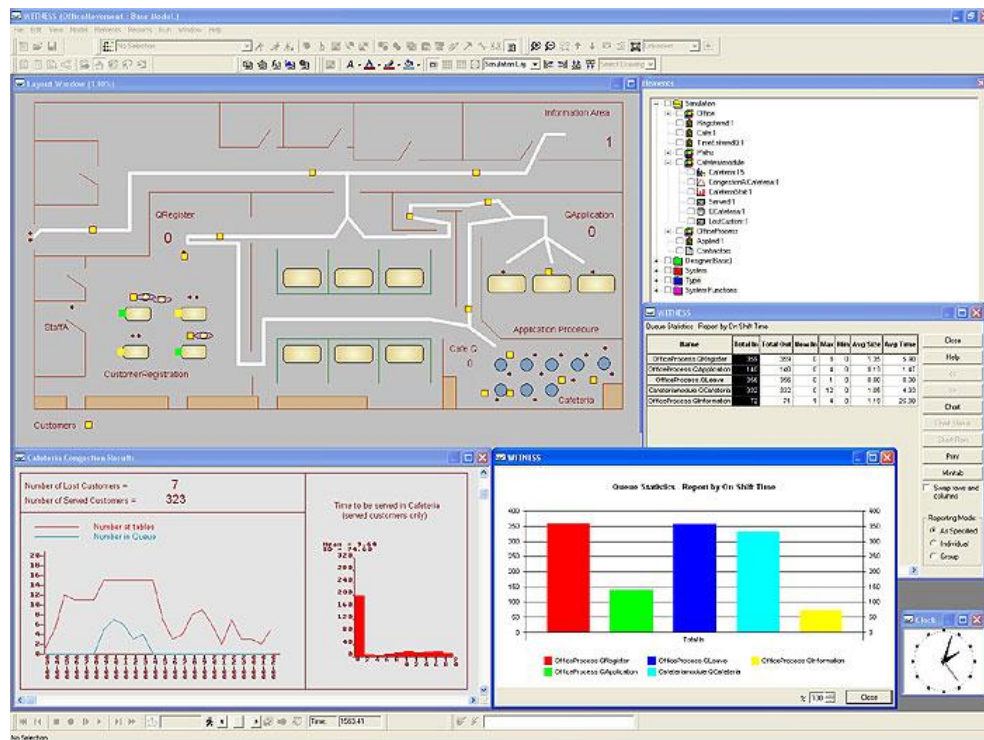
Es un software de simulación de eventos discretos, orientado a procesos logísticos y de fabricación, basado en el paradigma de construcción y programación orientada a objetos y modelado basado en la animación 3D. La última versión de WITNESS es la 12, es elaborado por la empresa Lanner Group Ltd.

Tabla 4. Witness.

 	
<b>Software</b>	Witness v. 12
<b>Empresa</b>	Lanner Group
<b>Clasificación</b>	Eventos Discretos y Continuos
<b>Módulos</b>	Witness Optimizer, Witness VR, Visio Link

Fuente: Lanner Group Ltd.

Figura 10. Modelo de distribución de planta en Witness.



Fuente: Sitio Web de Witness.

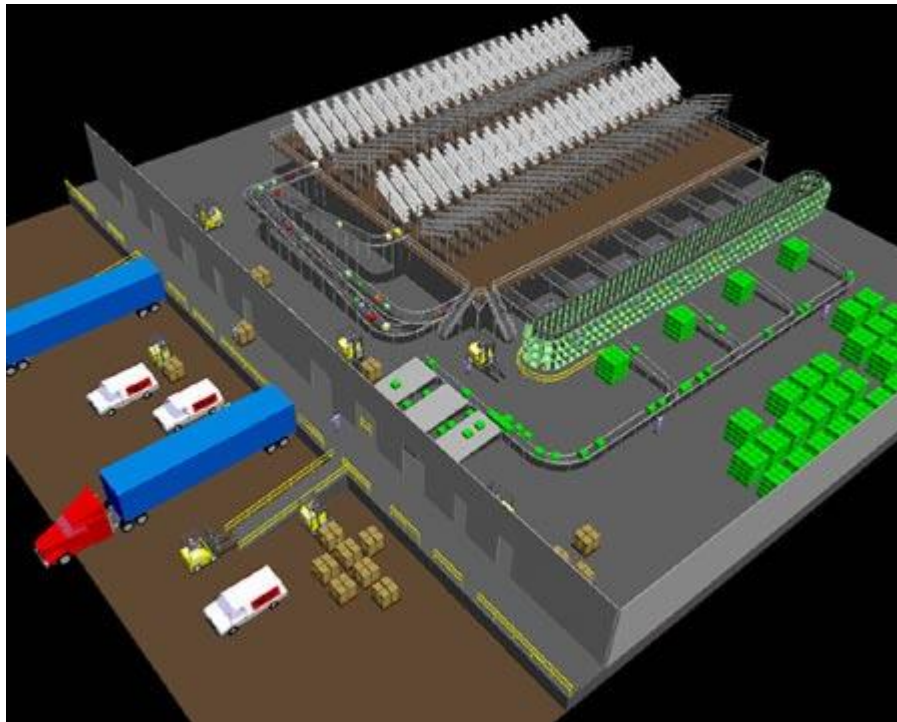
### 3.2.5 AUTOMOD

Desarrollado por Applied Materials. Es un software de simulación de eventos discretos y continuos basado en el paradigma de construcción y programación orientada a objetos y modelado basado en la animación (3D).

Tiene los siguientes módulos:

- ❖ **AutoMod VR.** Animación gráfica de realidad virtual que ayuda a validar el modelo y comunicar o presentar el diseño visualmente.
- ❖ **AutoStat (Experiments, Statistics).** Módulo para el diseño de experimentos y pruebas de bondad de ajuste de datos.

Figura 11. Centro de Distribución en AutoMod.



Fuente: AutoMod.

### **3.2.6 EXTENDSIM**

Desarrollado por la compañía “Imagine That!”, y completamente renovado en su última versión ExtendSim V7 logrando una herramienta potente con más precisión.

Está compuesta por los siguientes módulos:

- ❖ ExtendSim CP: Procesos continuos.
- ❖ ExtendSim OR: Investigación de operaciones.
- ❖ ExtendSim AT: Tecnología avanzada.
- ❖ ExtendSim Suite: Ambiente profesional 3D.

Este software de simulación sirve para modelar eventos discretos y continuos.

### **3.2.7 SIMUL8**

Este producto de SIMUL8 Corporation, es un software de simulación de eventos discretos y continuos basado en el paradigma de construcción y programación orientada a objetos y modelado basado en la animación (3D).

#### 4. INSTALACIÓN Y CONFIGURACIÓN DE FLEXSIM

La licencia académica de Flexsim que posee la Universidad Industrial de Santander es de tipo concurrente, es decir, hay un equipo servidor en la sala de cómputo de la EEIE que se conecta a un servidor de Flexsim en Utah, Estados Unidos para validar la licencia y los usuarios (seats) disponibles que pueden acceder simultáneamente al software, luego los equipos cliente o de los usuarios se conectan al gestor de licencias en el servidor dispuesto por la EEIE.

Por esta razón para la utilización completa de Flexsim, con todos sus módulos, es necesario siempre estar conectado a Internet.

La versión Trial o de prueba solo admite la construcción de 20 objetos, impidiendo acceder a algunas de las opciones más importantes de Flexsim, como es el uso de variables globales, el acceso al árbol de objetos y los principales módulos de este software.

La EEIE dispone de una licencia para 30 equipos cliente. La última versión licenciada es la 6.0.2.

En algunas ocasiones, Flexsim presenta problemas al instalarse porque necesita un archivo dll (biblioteca de vínculo dinámico) de SketchUp que al momento de la instalación puede estar corrupto. Para solucionarlo se debe instalar la versión gratuita o full (con licencia) de Google SketchUp<sup>17</sup>.

Para la administración del gestor de licencias de Flexsim, se desarrolló un protocolo de configuración e instalación de la licencia concurrente de Flexsim.

---

<sup>17</sup> Software de modelamiento en 3D de la Google Inc.

Este documento se entrega al administrador de la sala de cómputo de la Escuela de Estudios Industriales y Empresariales y al director del proyecto.

Por el diseño del documento mencionado y la protección e integridad de la licencia de tipo concurrente, no se anexa el documento en este trabajo de grado por ser de carácter confidencial , además este tipo de licencias son de carácter académico y empresarial, donde hay un equipo central o servidor que atiende las solicitudes de equipos cliente como usuarios del sistema, sean empleados, estudiantes o docentes, por lo cual difiere mucho de las licencias monousuario a las que suele estar acostumbrado la mayoría de usuarios.

#### **4.1 ACCESO REMOTO CON VPN**

Es posible acceder remotamente a Flexsim con una VPN o red privada virtual, es decir, por fuera de la LAN o la red local donde se encuentra el servidor. Esto es muy práctico cuando se requiere acceder a software licenciado y el docente o el estudiante no se encuentran dentro del campus universitario (pero está activo y matriculado en la asignatura donde se enseña Flexsim). Sobre todo la finalidad es que pueda seguir practicando y mejorando sus estrategias de aprendizaje, sin embargo, requerirá del permiso del administrador del sistema y de la Escuela de Estudios Industriales y Empresariales.

El uso de este tipo de VPNs requiere software comercial, que es muy económico, en el caso del software LogMeIn Hamachi, un año de servicio con capacidad para alojar a máximo 32 miembros por red está costando 29 dolares<sup>18</sup> a Marzo de 2013, por lo cual los beneficios obtenidos, al tener acceso en horas extra clase

---

<sup>18</sup> Consultado en: LogMeIn. Recuperado el 15 de marzo de 2013 en: <https://secure.logmein.com/US-ES/products/hamachi/purchase.aspx>

definidas previamente implica que se puedan desarrollar modelos más complejos y hacer un mejor uso de las TICs y los ambientes virtuales de aprendizaje (AVA).

Acceder de esta manera requerirá tener un horario establecido de atención del servidor para usuarios remotos para que no interfieran con los usuarios de la sala de cómputo y de la asignatura, por lo tanto se sugiere crear un protocolo de acceso remoto a estudiantes de la asignatura Técnicas Modernas de Optimización y trabajar en grupos de dos a tres estudiantes con un horario determinado de acceso, en caso de que la EEIE pague por la anualidad de una VPN.

Figura 12. VPN con el software Hamachi



Fuente: Interfaz LogMeIn Hamachi.

## 5. METODOLOGÍA DE ENSEÑANZA-APRENDIZAJE

La finalidad de la enseñanza se centra en la formación de estudiantes competentes que orienten y lideren un progreso intelectual, industrial y económico. La enseñanza como fuente, la didáctica como medio y el aprendizaje como receptor de la cadena, son los principales entes que se tendrán en cuenta para el desarrollo metodológico y aplicación de los talleres.

Figura 13. Estrategia de Enseñanza - Aprendizaje



Fuente: Corredor, et al. Estrategias de Enseñanza y Aprendizaje. CEDEUIS, Bucaramanga, 2009.

### 5.1 LAS TICS EN LOS PROCESOS DE ENSEÑANZA Y APRENDIZAJE

Las Tecnologías de la Información y la Comunicación (TICs) permiten una mayor interacción y comunicación entre las personas, y la posibilidad de compartir información que facilite la utilización de cierto tipo de aprendizaje como puede ser el colaborativo.

La pedagogía constructivista en la que se basa el aprendizaje colaborativo, sostiene que el conocimiento no se recibe de manera pasiva, sino que es construido activamente por el sujeto, transformando el rol del docente como un mediador entre el conocimiento y el alumno. Por lo tanto, no solamente es importante la participación, sino que es fundamental que en el proceso de enseñanza-aprendizaje, el sujeto que aprende tenga la posibilidad de autogestionar su propio aprendizaje.

Una de las herramientas tecnológicas a disposición de los docentes de la UIS, empleada en este proyecto de grado<sup>19</sup> y que promueve el aprendizaje con estas características es Moodle (Module Object-Oriented Dynamic Learning Environment- Entorno Modular de Aprendizaje Dinámico Orientado a Objetos), un sistema de gestión del aprendizaje (LMS - *Learning Management System*). Esta plataforma de software alojada en un servidor web y accesible desde Internet, se emplea para administrar, distribuir y controlar las actividades de formación no presencial y complementar la educación presencial, además permite que los estudiantes contribuyan en el aprendizaje mediante su participación en foros, blogs, comentarios de las actividades y a su vez el docente pueda realizar un seguimiento de los alumnos inscritos. Otra herramienta disponible es Faronics Insight, un software gestión de clases, donde el docente comparte la pantalla de su computador con sus estudiantes facilitando el proceso de enseñanza y puede guiarlos tomando control de sus terminales.

En los procesos de enseñanza es importante participar en la generación de esa cultura integrándola en la Educación, se deben usar las TIC para aprender y para enseñar. Es decir el aprendizaje de cualquier materia o habilidad se puede facilitar mediante las TIC y, en particular, mediante Internet, desarrollando sistemas de enseñanza que relacionen los distintos aspectos de la Informática y de la transmisión de información, siendo al mismo tiempo lo más constructivos que sea posible desde el punto de vista metodológico.

---

<sup>19</sup> Plataforma Moodle del director del proyecto y Portal Web del profesor Edwin Garavito - UIS: [http://gavilan.uis.edu.co/~garavito/index\\_general.htm](http://gavilan.uis.edu.co/~garavito/index_general.htm)

## 5.2 ESTRATEGIAS PARA LA ENSEÑANZA

La educación del manejo de nuevas tecnologías y aplicaciones de software tiene como principal reto la formación integral del estudiante, ofreciendo experiencias de enseñanza y aprendizaje que faciliten la formación de estudiantes partiendo desde sus conceptos básicos. Las diferentes estrategias de enseñanza expuestas por Corredor (2009) (ver tabla No. 5), permiten una clara visión de las posibles técnicas utilizadas en la enseñanza.

Tabla 5. Modalidades estrategias de enseñanza

<b>Criterio</b>	<b>Tipo</b>	<b>Finalidad</b>	<b>Modalidad</b>
<b>Según el fin que se persigue</b>	<b>Organizativas</b>	Interacción	Preguntas, trabajo en grupo, debate, panel, secuencia, foro, lluvia de ideas por tarjetas
		Comunicación	Narración, bola de nieve, simposio, diálogos simultáneos, debate
		Socialización	Trabajo en grupo, debate lluvia de ideas, reja foro, trabajo por relevos
		Cooperación	Proyectos, lectura comentada, rompecabezas, aprendizaje cooperativo
		Intercambio de experiencias	Grupos de discusión, foro, proyectos, quintetos de rotación
		Trabajo independiente	Lecturas, composiciones, problemas
<b>Según el fin que se persigue</b>	<b>Cognitivas</b>	Activación y generación de conocimiento previo	Preguntas, lluvia de ideas, discusión guiada, lluvia de ideas con tarjetas, objetivos
		Mantenimiento y guía de la atención	Preguntas, ilustraciones, objetivos, pistas tipográficas

criterio	Tipo	Finalidad	Modalidad
Según el fin que se persigue	Cognitivas	Organización de contenidos y procesos	Esquemas, resúmenes, clasificaciones, mapas conceptuales, organización de cursos de acción, redes semánticas, uve heurística
		Identificación de errores	Lectura comentada, problemas guiados, ensayos, ejercicios, proyectos dirigidos.
		Comprensión de contenidos y proceso	Mapas conceptuales, redes semánticas, mapas mentales, analogías, debates, ensayos, proyectos.
		Verificación de logros	Preguntas, debates, estudio de casos, problemas, proyectos
Según la etapa del proceso en que se usen	Motivación	Para lograr la implicación y participación activa del estudiante en proceso	Correlación de los contenidos estudiados con lo real, el uso de elogios y censuras, la utilización de material didáctico, la organización del ambiente, el conocimiento de los objetivos y propósitos de la tarea, la personalidad del docente, la realización de experiencias donde se aplique o vea la aplicación de lo aprendido
	Elaboración	Favorecen el procesamiento de los estímulos que provienen de fuentes externas e internas y con los cuales se trabaja para la construcción de conocimientos y desarrollo de competencias	<ul style="list-style-type: none"> <li>✓ <u>De comunicación directa</u>: exposición dogmática, exposición dialogada, lectura en todas las modalidades, demostración.</li> <li>✓ <u>De aprendizaje individual</u>: fichas, guías de estudio, módulos de autoinstrucción.</li> <li>✓ <u>De aprendizaje en grupo</u>: discusión, debate, círculo de estudio, panel, mesa redonda, foro, estudio de casos, lluvia de ideas, juego de roles.</li> </ul>

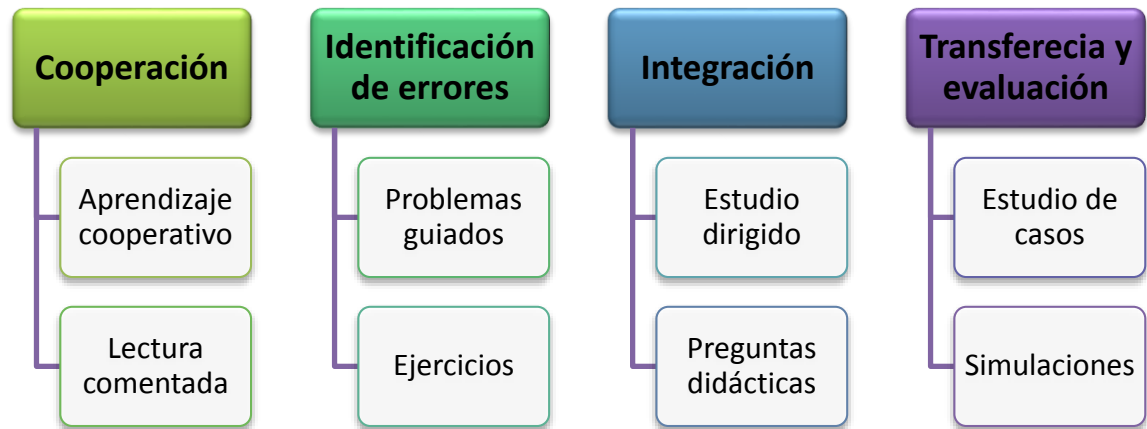
<b>Criterio</b>	<b>Tipo</b>	<b>Finalidad</b>	<b>Modalidad</b>
<b>Según la etapa del proceso en que se usen</b>	<b>Elaboración</b>		✓ <u>De formación para la investigación:</u> proyecto, laboratorio, problemas, trabajo de campo, búsqueda bibliográfica, informe científico, monografía, simulación de procesos, pasantías, uve heurística.
	<b>Fijación</b>	Para la apropiación de conceptos, procedimientos, habilidades, actitudes y valores	Ejercicios y tareas, trabajo práctico, interrogatorio, resumen, mapas conceptuales, esquemas, redes semánticas, mapas mentales, guías didácticas, cuadros sinópticos, estudio de casos, debate, foro, estudio dirigido, problemas
	<b>Integración</b>	Para la integración de conceptos y experiencias a las estructuras conceptuales previas de los estudiantes	Preguntas didácticas, debate, estudio dirigido, torneos orales y escritos, mapas conceptuales, mapas mentales, redes semánticas, esquemas, resúmenes, áreas dirigidas
	<b>Transferencia y evaluación</b>	Para la valoración y aplicación de lo aprendido en situaciones y problemas diferentes a los que se dieron en el momento del aprendizaje	Estudio de casos, proyectos, trabajo de campo, problemas, toma de decisiones, investigación, diseño, simulaciones, juego de roles

Fuente: CORREDOR, et al. (2009), pág. 65-72

La implementación de los talleres requiere una percepción teórica previa, herramientas TICs y competencia sobre el software del exponente, luego es pertinente hacer grupos de trabajo de dos estudiantes para fomentar la integración y el aprendizaje colaborativo entre ellos mismos, donde se manifestará las

habilidades de cada uno y el trabajo cooperativo; también las tutorías por parte del expositor para acompañar el aprendizaje del estudiante y así involucrar las estrategias de enseñanza<sup>20</sup> presentadas en la Figura No. 14:

Figura 14. Estrategias de enseñanza involucradas en los talleres



Fuente: CORREDOR, et al. (2009), pág. 65-72.

### 5.3 ESTRATEGIA DIDÁCTICA

Para la enseñanza del software Flexsim, se diseñó como objeto de aprendizaje un instructivo o manual como estrategia principal, concebida como herramienta práctica de trabajo, guía de referencia y ayuda didáctica. Este instructivo se acompaña de unos talleres prácticos para desarrollar en clase y aplicar los conocimientos adquiridos, siguiendo un enfoque constructivista, cada taller repasa elementos vistos en los talleres previos y de esta manera al desarrollar el último taller puedan integrar todos los elementos aprendidos.

<sup>20</sup> Ver: Corredor, et al. 2009, p. 65-72.

Asimismo, para una mejor asistencia en el aprendizaje colaborativo, se apoya el direccionamiento de los talleres por medio del software de gestión de clases Faronics Insight<sup>21</sup> y de los videos explicativos que acompañan los talleres.

### **5.3.1 Diseño Instruccional del Objeto de Aprendizaje**

El diseño instruccional hace referencia a un proceso planificado con una estructura definida de creación de materiales u objetos necesarios para lograr un aprendizaje.

El objeto de aprendizaje sigue este proceso secuencial para cumplir los siguientes objetivos:

- ❖ Ser una guía de referencia para el estudiante.
  
- ❖ Presentar talleres que ayuden a afianzar los conocimientos adquiridos en clase.

Para la implementación del objeto de aprendizaje se utilizó material didáctico como presentaciones y algunos vídeos que acompañan el instructivo desarrollado.

## **5.4 ESTRATEGIAS PARA EL APRENDIZAJE**

Las tecnologías de la información, exigen cada vez más un aprendizaje eficiente para el desarrollo integral. La tabla No. 6 presenta las diferentes metodologías de

---

<sup>21</sup> Faronics Insight: Software que ayuda a los instructores a proporcionar una mejor experiencia de aprendizaje con la posibilidad de educar, asistir, supervisar y comunicarse con toda la clase desde un computador central que tiene la capacidad de guiar al estudiante remotamente.

aprendizaje frente a las estrategias o técnicas que se pueden desarrollar en el taller.

Tabla 6. Clasificación de estrategias y técnicas según la participación

<b>Participación</b>	<b>Ejemplos de estrategias y técnicas (Actividades)</b>
<b>Autoaprendizaje</b>	<ul style="list-style-type: none"> <li>• Estudio individual</li> <li>• Búsqueda y análisis de información</li> <li>• Elaboración de ensayos</li> <li>• Tareas individuales</li> <li>• Proyectos</li> <li>• Investigaciones</li> </ul>
<b>Aprendizaje interactivo</b>	<ul style="list-style-type: none"> <li>• Exposiciones del profesor</li> <li>• Conferencia con un experto</li> <li>• Entrevistas</li> <li>• Visitas</li> <li>• Paneles</li> <li>• Debates</li> <li>• Seminarios</li> </ul>
<b>Aprendizaje colaborativo</b>	<ul style="list-style-type: none"> <li>• Solución de casos</li> <li>• Métodos de proyectos</li> <li>• Aprendizaje basado en problemas</li> <li>• Análisis y discusión de grupos</li> <li>• Discusión y debates</li> </ul>

Fuente:

[http://acreditacion.udistrital.edu.co/flexibilidad/estrategias\\_didacticas\\_aprendizaje\\_colaborativo.pdf](http://acreditacion.udistrital.edu.co/flexibilidad/estrategias_didacticas_aprendizaje_colaborativo.pdf)

### 5.4.1 Características del Aprendizaje Colaborativo

Según Fabra (1994), los elementos presentes en este tipo de aprendizaje son:

- ❖ **Cooperación:** Los estudiantes se apoyan mutuamente para cumplir con un doble objetivo: lograr ser expertos en el conocimiento del contenido, además de desarrollar habilidades de trabajo en equipo. Los estudiantes comparten metas, recursos, logros y entendimiento del rol de cada uno. Un estudiante no puede tener éxito a menos que todos en el equipo tengan éxito.
- ❖ **Responsabilidad:** Los estudiantes son responsables de manera individual de la parte de tarea que les corresponde. Al mismo tiempo, todos en el equipo deben comprender todas las tareas que les corresponden a los compañeros.
- ❖ **Comunicación:** Los miembros del equipo intercambian información importante y materiales, se ayudan mutuamente de forma eficiente y efectiva, ofrecen retroalimentación para mejorar su desempeño en el futuro y analizan las conclusiones y reflexiones de cada uno para lograr pensamientos y resultados de mayor calidad.
- ❖ **Trabajo en equipo:** Los estudiantes aprenden a resolver juntos los problemas, desarrollando las habilidades de liderazgo, comunicación, confianza, toma de decisiones y solución de conflictos.
- ❖ **Autoevaluación:** Los equipos deben evaluar cuáles acciones han sido útiles y cuáles no. Los miembros de los equipos establecen las metas, evalúan periódicamente sus actividades e identifican los cambios que deben realizarse para mejorar su trabajo en el futuro.

## **6. INSTRUCTIVO DE APRENDIZAJE DEL SOFTWARE DE SIMULACIÓN FLEXSIM**

La elaboración del Instructivo de Aprendizaje del Software de Simulación Flexsim se le atribuye a la EEIE como una iniciativa para contribuir en la enseñanza de una herramienta poderosa de simulación con el propósito de modelar, analizar, visualizar y optimizar cualquier proceso industrial y de servicios, desde manufactura hasta cadena de abastecimiento.

### **6.1 ¿QUE ES UN INSTRUCTIVO Y/O MANUAL?**

Es un herramienta que contiene de manera explícita, ordenada y sistemática información de una organización o de un sistema como: objetivos, políticas, sistemas, procedimientos, funciones, atribuciones, antecedentes etc... Así como una serie de explicaciones e instrucciones que se consideren necesarias para darle a una persona la posibilidad de actuar de acuerdo a como sea requerido según la situación.<sup>22</sup>

### **6.2 OBJETIVOS DE UN MANUAL**

Uno de los principales objetivos con los que se desarrolla cualquier tipo de manual y/o instructivo es el de permitir a su usuario lograr llevar a cabo determinadas acciones de la mejor manera posible.

---

<sup>22</sup> Tomado de: definicion.org. Recuperado el 10 de agosto de 2012 en <http://www.definicion.org/manual/>

Entre los objetivos y beneficios de la elaboración de un Manual se destacan:

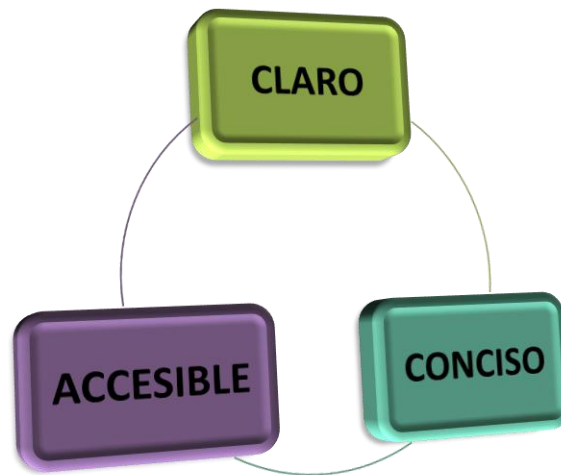
- ❖ Sirve para el análisis o revisión de los procedimientos de un sistema.
- ❖ Proporcionar información básica para la planeación e implementación de reformas administrativas.
- ❖ Interviene en la consulta de todo el personal.
- ❖ Servir de acompañamiento y apoyo en el proceso de aprendizaje.
- ❖ Mostrar requerimientos mínimos de cómo utilizar la herramienta.

### **6.3 CARACTERISTICAS**

Para obtener aquellos resultados esperados, el instructivo debe contar con algunas características básicas que faciliten la acción en sí. Entre estas características se puede mencionar la importancia de que sea claro y conciso. Es preciso que las instrucciones sean dadas de manera accesible de modo que el usuario que las lee o sigue pueda comprenderlas fácilmente. En muchos casos, los instructivos pueden sumar imágenes y otros elementos para ayudar a la comprensión.

En conclusión, los instructivos no deben ser demasiado complejos y extensos ya que se pueden volver confusos y hacer que los usuarios se pierdan en el procedimiento.

Figura 15. Características del instructivo de aprendizaje



Fuente: Autor.

## 6.4 DISEÑO

### 6.4.1 ¿A quién va dirigido este Instructivo de Aprendizaje?

Este Instructivo va dirigido a cualquiera que desee plantear una simulación. Así mismo, tiene en cuenta el hecho de que las personas no necesariamente deben ser expertas en programación generando un mayor interés por parte de los estudiantes en los modelos a desarrollar.

Se espera que el presente instructivo también sea útil para los profesores y administradores de otras instituciones y para todos los niveles en la educación.

## 6.4.2 Estructura del instructivo de aprendizaje

El contenido del instructivo de aprendizaje incluye el diseño mostrado a continuación, para facilitar su enfoque metodológico.

- ❖ Capítulo 1, Acerca de Flexsim: contiene una breve descripción del software, de las aplicaciones industriales y de servicios, se describen sus módulos y componentes.
- ❖ Capítulo 2, Interfaz Gráfica de Flexsim: se explica todo el entorno gráfico, las barras de herramientas y se hace un recorrido por cada uno de los menús de Flexsim. Se introduce la librería de objetos y se describen cada uno de ellos. Por último se explican las vistas del modelo.
- ❖ Capítulo 3, Construcción de modelos: se explica la terminología empleada en simulación y el argot de Flexsim. Se crean los primeros objetos y hace énfasis en las propiedades de estos y sus múltiples maneras de edición.
- ❖ Capítulo 4, Adicionando la lógica del modelo: en este capítulo se define el flujo del modelo, las características inherentes a ellos y se realiza la introducción de elementos adicionales para la simulación, como: las redes de transporte, las secuencias de tareas y las tablas globales.
- ❖ Capítulo 5, Utilización de las herramientas de Flexsim: introducción a las herramientas de análisis de datos.

## 7. TALLERES PRÁCTICOS DE SIMULACIÓN CON FLEXSIM

### 7.1 PLANEACIÓN Y DESARROLLO

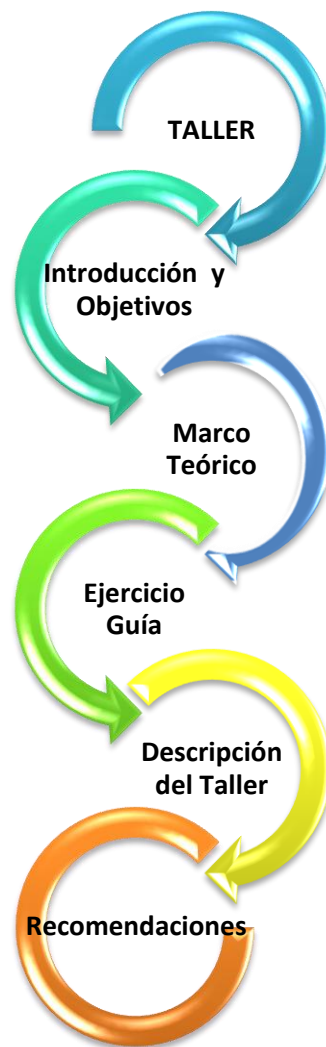
El manual de prácticas está diseñado por una serie de talleres guiados, el cual comienza con un introductorio que contiene los aspectos más relevantes del software, seguidamente aborda 6 talleres que definen una guía básica de conceptos teóricos, técnicos y un ejercicio de aplicación.

Cada taller se compone de un ejercicio guiado por parte del docente o expositor y de un taller propuesto para desarrollar en clase y dar cumplimiento al objetivo de la práctica. La figura No. 2 muestra la forma como se estructuró cada taller.

- ❖ **Introducción y Objetivos:** expone al estudiante una visión global de lo que desarrollará y cuáles son sus alcances, permitiendo una visualización de los temas teóricos, alcance y requisitos que debe reunir para implementarlo.
- ❖ **Marco Teórico:** muestra la teoría que se necesita e involucra conceptos para la interpretación del problema.
- ❖ **Ejercicio Guía:** presenta un ejercicio dirigido, desarrollado paso a paso para la correcta utilización del software. Es acompañado de un video de construcción parcial o total del modelo, además de las ilustraciones del taller.
- ❖ **Descripción del Taller:** describe los elementos a emplear, conceptos técnicos y teóricos frente al software de simulación Flexsim necesarios para la construcción y desarrollo del problema.

- ❖ **Recomendaciones:** se presentan sugerencias de construcción del modelo, posibles mejoras en la manipulación de variables o entidades, tópicos especiales del software y otras o mejores maneras de cumplir los objetivos del taller.
- ❖ **Hoja de Datos:** hoja con los datos de salida recopilados para su calificación por parte del docente.

Figura 16. Diseño de los talleres



Fuente: Autor.

## 7.2 TALLERES DESARROLLADOS EN LA ASIGNATURA.

Para realizar la prueba piloto se planeó e implementó una clase introductoria y posteriormente 3 talleres prácticos que se realizaron en el espacio práctico que tiene la asignatura Técnicas Modernas de Optimización. En las siguientes tablas, se muestran los seis talleres desarrollados para el presente proyecto, de los cuales se implementaron el taller tres, cuatro y cinco.

Tabla 7. Taller Introductorio

<b>TALLER INTRODUCTORIO</b>	
<b>DIRIGIDO POR</b>	Edwin Alberto Garavito Hernández
<b>TEMA</b>	<i>Introducción al Software Flexsim</i>
<b>FECHA DE REALIZACIÓN</b>	
<b>DURACIÓN</b>	2 horas
<b>OBJETIVO GENERAL</b>	
Introducir a los estudiantes de Técnicas Modernas de Optimización a la simulación de eventos discretos mediante software de última generación.	
<b>DESCRIPCIÓN</b>	
En el taller introductorio los estudiantes conocerán los conceptos generales de la simulación, la terminología o argot de Flexsim, el por qué es importante su uso en los problemas de la industria, así como una introducción al software Flexsim.	
<b>ACTIVIDADES A REALIZAR</b>	
<ul style="list-style-type: none"> <li>❖ Charla sobre simulación</li> <li>❖ Introducción y uso del software FLEXSIM</li> <li>❖ Interfaz Gráfica de Usuario (GUI)</li> <li>❖ Barra de herramientas (Toolbar)</li> <li>❖ Barra de menú</li> <li>❖ Biblioteca de objetos</li> <li>❖ Vistas del modelo</li> <li>❖ Inserción de objetos</li> <li>❖ Conexión de puertos</li> <li>❖ Taller básico de las herramientas principales del software FLEXSIM</li> </ul>	

Tabla 7. (Continuación)

<b>TALLER INTRODUCTORIO</b>	
<b>ESTRATEGIAS A UTILIZAR</b>	
<ul style="list-style-type: none"> <li>❖ Interacción</li> <li>❖ Cooperación</li> <li>❖ Socialización</li> <li>❖ Intercambio de experiencias</li> </ul>	<ul style="list-style-type: none"> <li>❖ Trabajo independiente</li> <li>❖ Identificación de errores</li> <li>❖ Comprensión de contenidos</li> <li>❖ Comprensión de proceso</li> </ul>
<b>RECURSOS NECESARIOS</b>	
<ul style="list-style-type: none"> <li>❖ 1 Computador para el docente</li> <li>❖ 1 Computador por estudiante</li> <li>❖ 1 Video beam</li> </ul>	<ul style="list-style-type: none"> <li>❖ Software                             <ul style="list-style-type: none"> <li>➤ Flexsim</li> <li>➤ Google SketchUp</li> <li>➤ Faronics Insight</li> </ul> </li> </ul>

Fuente: Autor.

Tabla 8. Taller No. 1

<b>TALLER No. 1</b>	
<b>DIRIGIDO POR</b>	Edwin Alberto Garavito Hernández
<b>TEMA</b>	<i>Simulación de líneas de espera</i>
<b>FECHA DE REALIZACIÓN</b>	
<b>DURACIÓN</b>	2 horas
<b>OBJETIVO GENERAL</b>	
Conocer y aplicar las herramientas del software Flexsim para la solución de eventos relacionados con la teoría de colas.	
<b>DESCRIPCIÓN</b>	
El primer taller permite al estudiante profundizar los conocimientos sobre el problema de teorías de colas, iniciado en el curso de Investigación de Operaciones II, esta vez accederá a situaciones más reales de la vida cotidiana, reflejando como producto final la solución de un problema guiado de simulación utilizando objetos básicos del software Flexsim como Fuentes, Filas, Estaciones y/o Disposición de entidades.	
<b>ACTIVIDADES A REALIZAR</b>	
<ul style="list-style-type: none"> <li>❖ Charla sobre la aplicación de la teoría de colas en la simulación</li> <li>❖ Explicar los conceptos y funciones de las herramientas a utilizar en el taller</li> <li>❖ Definir la configuración del problema</li> <li>❖ Solución de un problema guiado básico</li> <li>❖ Conclusión grupal de la experiencia en el desarrollo del problema</li> </ul>	
<b>ESTRATEGIAS A UTILIZAR</b>	
<ul style="list-style-type: none"> <li>❖ Interacción</li> <li>❖ Socialización</li> <li>❖ Intercambio de experiencias</li> </ul>	<ul style="list-style-type: none"> <li>❖ Trabajo independiente</li> <li>❖ Identificación de errores</li> <li>❖ Comprensión de contenidos y proceso</li> </ul>
<b>RECURSOS NECESARIOS</b>	
<ul style="list-style-type: none"> <li>❖ 1 Computador para el docente</li> <li>❖ 1 Computador por estudiante</li> <li>❖ 1 Video beam</li> </ul>	<ul style="list-style-type: none"> <li>❖ Software                             <ul style="list-style-type: none"> <li>➤ Flexsim</li> <li>➤ Faronics Insight</li> </ul> </li> </ul>

Fuente: Autor.

Tabla 9. Taller No. 2

<b>TALLER No. 2</b>	
<b>DIRIGIDO POR</b>	Edwin Alberto Garavito Hernández
<b>TEMA</b>	<i>Filas de espera con Tablas Globales</i>
<b>FECHA DE REALIZACIÓN</b>	
<b>DURACIÓN</b>	4 horas
<b>OBJETIVO GENERAL</b>	
Desarrollar un modelo de colas en Flexsim mediante la utilización de tablas globales para tiempos entre llegadas y tiempos de operación observados y tiempos que siguen una distribución empírica	
<b>DESCRIPCIÓN</b>	
El segundo taller continúa con el tema de filas de espera, utilizando tablas globales para modelar tiempos observados en Flexsim y contrastarlos con resultados teóricos obtenidos en Excel.	
<b>ACTIVIDADES A REALIZAR</b>	
<ul style="list-style-type: none"> <li>❖ Explicación de la utilización de tablas globales</li> <li>❖ Presentación del diagrama de flujo del problema</li> <li>❖ Modelado del proceso en con el software</li> <li>❖ Verificación de la solución de cada grupo mediante el software de control remoto Faronics Insight</li> </ul>	
<b>ESTRATEGIAS A UTILIZAR</b>	
<ul style="list-style-type: none"> <li>❖ Interrogativas, Interacción</li> <li>❖ Demostración, Cooperación</li> <li>❖ Socialización, Ilustraciones</li> <li>❖ Intercambio de experiencias</li> </ul>	<ul style="list-style-type: none"> <li>❖ Trabajo independiente</li> <li>❖ Identificación de errores</li> <li>❖ Comprensión de contenidos</li> <li>❖ Comprensión de proceso</li> </ul>
<b>RECURSOS NECESARIOS</b>	
<ul style="list-style-type: none"> <li>❖ 1 Computador (docente)</li> <li>❖ 1 Computador por estudiante</li> <li>❖ 1 Video beam</li> </ul>	<ul style="list-style-type: none"> <li>❖ Software                             <ul style="list-style-type: none"> <li>➤ Flexsim</li> <li>➤ Faronics Insight</li> </ul> </li> </ul>

Fuente: Autor.

Tabla 10. Taller No. 3

<b>TALLER No. 3</b>	
<b>DIRIGIDO POR</b>	William Jesús Pinto Triana
<b>TEMA</b>	<i>Análisis de Datos con ExpertFit</i>
<b>FECHA DE REALIZACIÓN</b>	09 / 04 / 2012
<b>DURACIÓN</b>	4 horas
<b>OBJETIVO GENERAL</b>	
Determinar la distribución de probabilidad a partir de un conjunto de datos utilizando la herramienta o módulo ExpertFit del software Flexsim.	
<b>DESCRIPCIÓN</b>	
En el tercer taller los estudiantes pondrán en práctica sus conocimientos teóricos de pruebas de bondad de ajuste para ajustar un conjunto de datos a una distribución teórica de probabilidad conocida o a una distribución empírica, así como los parámetros que componen determinadas distribuciones, además, conocerá las herramientas que posee el software Flexsim para el análisis de datos de entrada.	
<b>ACTIVIDADES A REALIZAR</b>	
<ul style="list-style-type: none"> <li>❖ Charla sobre fuentes de datos y distribuciones de probabilidad</li> <li>❖ Alistamiento de datos en Excel</li> <li>❖ Importación en ExpertFit de los datos de Excel</li> <li>❖ Análisis de datos con el módulo ExpertFit de Flexsim</li> <li>❖ Desarrollo del problema propuesto en el taller</li> <li>❖ Retroalimentación de la importancia de los datos en la simulación</li> </ul>	
<b>ESTRATEGIAS A UTILIZAR</b>	
<ul style="list-style-type: none"> <li>❖ Interacción</li> <li>❖ Cooperación</li> <li>❖ Socialización</li> <li>❖ Intercambio de experiencias</li> </ul>	<ul style="list-style-type: none"> <li>❖ Trabajo independiente</li> <li>❖ Identificación de errores</li> <li>❖ Comprensión de contenidos</li> <li>❖ Comprensión de proceso</li> </ul>
<b>RECURSOS NECESARIOS</b>	
<ul style="list-style-type: none"> <li>❖ 1 Computador (docente)</li> <li>❖ 1 Computador por estudiante</li> <li>❖ 1 Video beam</li> </ul>	<ul style="list-style-type: none"> <li>❖ Software                             <ul style="list-style-type: none"> <li>➤ Flexsim</li> <li>➤ ExpertFit</li> <li>➤ Microsoft Excel</li> <li>➤ Faronics Insight</li> </ul> </li> </ul>

Fuente: Autor.

Tabla 11. Taller No. 4

<b>TALLER No. 4</b>	
<b>DIRIGIDO POR</b>	William Jesús Pinto Triana
<b>TEMA</b>	<b><i>Simulación de Escenarios (Experimenter)</i></b>
<b>FECHA DE REALIZACIÓN</b>	18 / 04 / 2012
<b>DURACIÓN</b>	4 horas
<b>OBJETIVO GENERAL</b>	
Experimentar los posibles escenarios que pueda adoptar la solución de un sistema mediante el módulo Experimenter de Flexsim.	
<b>DESCRIPCIÓN</b>	
<p>El objeto de la simulación es precisamente ese, experimentar con los diferentes recursos que se tengan o se planeen adquirir, por otra parte la toma de decisiones involucra dos o más alternativas.</p> <p>El software Flexsim brinda una herramienta para analizar los diferentes escenarios que se puedan presentar en la toma de una decisión según su objetivo, así los estudiantes se instruyen en la herramienta Experimenter y cómo es su funcionamiento dentro de los parámetros de determinado problema.</p>	
<b>ACTIVIDADES A REALIZAR</b>	
<ul style="list-style-type: none"> <li>❖ Explicación del módulo Experimenter del software Flexsim</li> <li>❖ Definición de los escenarios de simulación.</li> <li>❖ Desarrollo del problema propuesto en el taller</li> <li>❖ Conclusiones del taller y ventajas de los escenarios en la toma de decisiones de los procesos de producción</li> </ul>	
<b>ESTRATEGIAS A UTILIZAR</b>	
<ul style="list-style-type: none"> <li>❖ Interacción</li> <li>❖ Cooperación</li> <li>❖ Socialización</li> <li>❖ Intercambio de experiencias</li> </ul>	<ul style="list-style-type: none"> <li>❖ Trabajo independiente</li> <li>❖ Identificación de errores</li> <li>❖ Comprensión de contenidos</li> <li>❖ Comprensión de proceso</li> </ul>
<b>RECURSOS NECESARIOS</b>	
<ul style="list-style-type: none"> <li>❖ 1 Computador (docente)</li> <li>❖ 1 Computador por estudiante</li> <li>❖ 1 Video beam</li> </ul>	<ul style="list-style-type: none"> <li>❖ Software                             <ul style="list-style-type: none"> <li>➤ Flexsim</li> <li>➤ Faronics Insight</li> </ul> </li> </ul>

Fuente: Autor.

Tabla 12. Taller No. 5

<b>TALLER No. 5</b>	
<b>DIRIGIDO POR</b>	William Jesús Pinto Triana
<b>TEMA</b>	<b><i>Sistemas de Control de Producción</i></b>
<b>FECHA DE REALIZACIÓN</b>	26 / 04 / 2012
<b>DURACIÓN</b>	4 horas
<b>OBJETIVO GENERAL</b>	
<p>Simular un sistema de producción CONWIP (inventario en proceso constante) vs. un sistema Push, y comparar las medidas de desempeño del sistema.</p>	
<b>DESCRIPCIÓN</b>	
<p>Uno de los objetivos más importantes a nivel operacional en los sistemas de producción es tener la cantidad mínima de inventario en proceso, por lo cual, el taller comprende un problema en el que el estudiante debe simular un proceso productivo mediante el sistema Push, Pull y el sistema Conwip, aplicando conceptos teóricos pertinentes a este caso en particular.</p>	
<b>ACTIVIDADES A REALIZAR</b>	
<ul style="list-style-type: none"> <li>❖ Charla sobre los sistemas de control de producción e inventarios</li> <li>❖ Introducción a los Sistemas híbridos Push-Pull</li> <li>❖ Descripción del proceso productivo</li> <li>❖ Simulación del proceso productivo mediante el sistema Push</li> <li>❖ Simulación del proceso productivo mediante el sistema Pull</li> <li>❖ Simulación del proceso productivo mediante el sistema Conwip</li> <li>❖ Análisis de estadísticas</li> <li>❖ Comparativa de los 3 sistemas</li> </ul>	
<b>ESTRATEGIAS A UTILIZAR</b>	
<ul style="list-style-type: none"> <li>❖ Interacción</li> <li>❖ Cooperación</li> <li>❖ Socialización</li> <li>❖ Intercambio de experiencias</li> </ul>	<ul style="list-style-type: none"> <li>❖ Trabajo independiente</li> <li>❖ Identificación de errores</li> <li>❖ Comprensión de contenidos</li> <li>❖ Comprensión de proceso</li> </ul>
<b>RECURSOS NECESARIOS</b>	
<ul style="list-style-type: none"> <li>❖ 1 Computador (docente)</li> <li>❖ 1 Computador por estudiante</li> <li>❖ 1 Video beam</li> </ul>	<ul style="list-style-type: none"> <li>❖ Software                             <ul style="list-style-type: none"> <li>➤ Flexsim</li> <li>➤ Faronics Insight</li> </ul> </li> </ul>

Fuente: Autor.

Tabla 13. Taller No. 6

<b>TALLER No. 6</b>	
<b>DIRIGIDO POR</b>	
<b>TEMA</b>	<i>Taller Integrador con Programación</i>
<b>FECHA DE REALIZACIÓN</b>	
<b>DURACIÓN</b>	8 horas
<b>OBJETIVO GENERAL</b>	
Construir un modelo que contenga todos los elementos de simulación más relevantes descritos y desarrollados en los talleres anteriores.	
<b>DESCRIPCIÓN</b>	
<p>El desarrollo del taller final trata de un ejercicio práctico que desarrollaran los estudiantes y que abarque la mayoría de herramientas del software Flexsim desarrolladas en los talleres anteriores, así como: ExpertFit, Experimenter, y utilización de elementos como: atributos, variables globales, eventos de usuario, etc.</p> <p>Adicionalmente se propone una parte de programación en código Flexscript para exponer la opción integrada al software Flexsim.</p>	
<b>ACTIVIDADES A REALIZAR</b>	
<ul style="list-style-type: none"> <li>❖ Retroalimentación de los talleres anteriores</li> <li>❖ Presentación de la práctica</li> <li>❖ Programación en código Flexscript de algunas necesidades específicas para el modelamiento en Flexsim y solución del problema</li> <li>❖ Supervisión y seguimiento de las entregas parciales y final de cada grupo.</li> <li>❖ Asistencia a través del software de control remoto Faronics Insight</li> </ul>	
<b>ESTRATEGIAS A UTILIZAR</b>	
<ul style="list-style-type: none"> <li>❖ Interacción</li> <li>❖ Cooperación</li> <li>❖ Socialización</li> <li>❖ Intercambio de experiencias</li> </ul>	<ul style="list-style-type: none"> <li>❖ Trabajo independiente</li> <li>❖ Identificación de errores</li> <li>❖ Comprensión de contenidos</li> <li>❖ Comprensión de proceso</li> </ul>
<b>RECURSOS NECESARIOS</b>	
<ul style="list-style-type: none"> <li>❖ 1 Computador (docente)</li> <li>❖ 1 Computador por estudiante</li> <li>❖ 1 Video beam</li> </ul>	<ul style="list-style-type: none"> <li>❖ Software                             <ul style="list-style-type: none"> <li>➤ Flexsim</li> <li>➤ Faronics Insight</li> </ul> </li> </ul>

Fuente: Autor.

### 7.3 EVALUACIÓN DE LA PRUEBA PILOTO

Para la evaluación de la prueba piloto se realizó una encuesta donde la calificación otorgada está compuesta por los siguientes niveles de satisfacción:

1. Deficiente	2. Regular	3. Bueno	4. Muy bueno	5. Excelente
---------------	------------	----------	--------------	--------------

La encuesta está compuesta por cuatro componentes que de manera general abarcan elementos de la estrategia de enseñanza, pedagógica y de aprendizaje, los cuales son:

Figura 17. Componentes de la evaluación de la prueba piloto



Fuente: Autor.

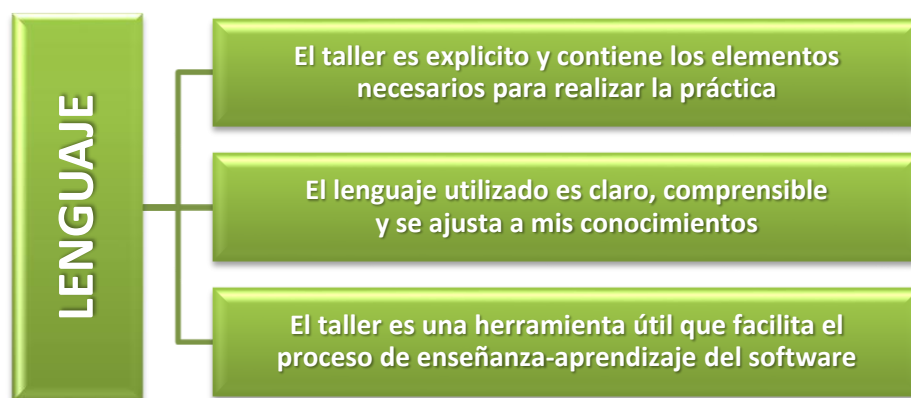
Los componentes: *Presentación*, *Lenguaje*, *Cobertura y Profundidad* y finalmente *Utilidad* tienen varias afirmaciones, las cuales se relacionan a continuación.

Figura 18. Componente de la encuesta: Presentación



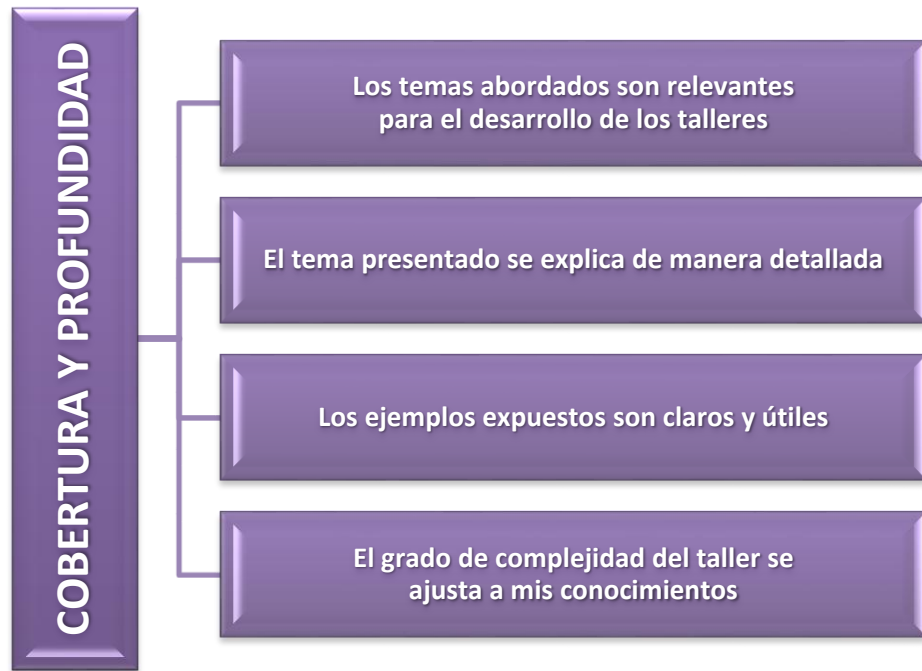
Fuente: Autor.

Figura 19. Componente de la encuesta: Lenguaje



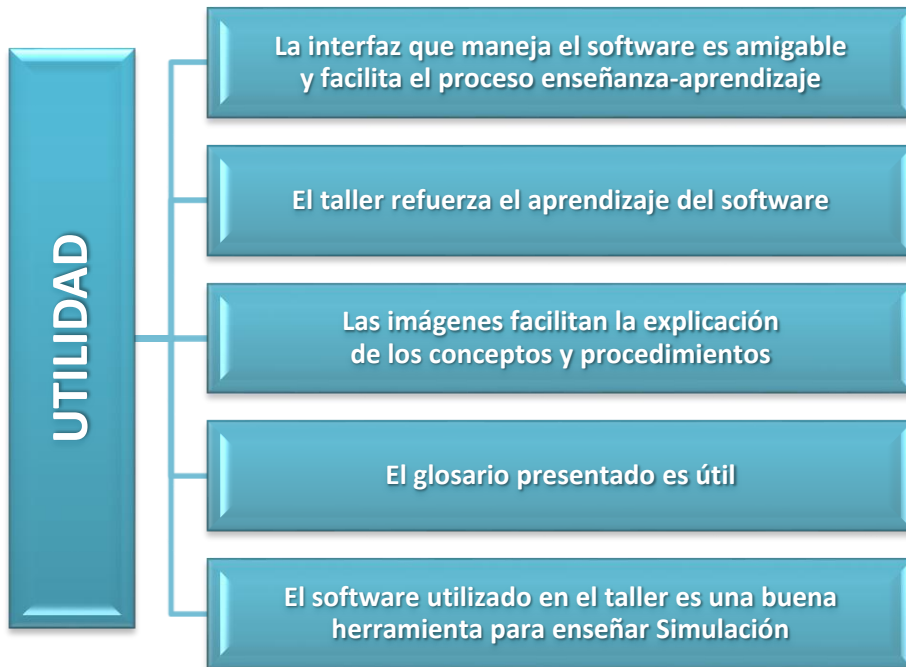
Fuente: Autor.

Figura 20. Componente de la encuesta: Cobertura y Profundidad



Fuente: Autor.

Figura 21. Componente de la encuesta: Utilidad



Fuente: Autor.

El modelo de la encuesta diseñada se encuentra en el Anexo 4 donde permite observar la relación de los diferentes componentes con sus afirmaciones vs los niveles de satisfacción de los estudiantes asistentes a los talleres piloto.

La encuesta se aplica a nivel de los 3 talleres implementados en el aula (talleres 3, 4 y 5 del manual), para efectos estadísticos la población de estudiantes asistentes en los 3 talleres es de 10 estudiantes, por lo tanto la muestra poblacional es el total de la población, a continuación se presenta los resultados de la encuesta.

### 7.3.1 Presentación

**P1.** ¿La presentación general del material facilita su lectura y utilización como guía de estudio?

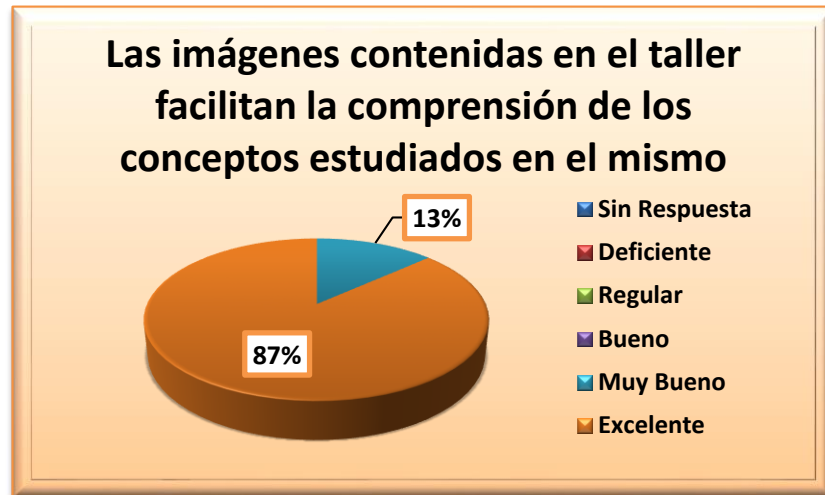
Figura 22. Resultados pregunta 1 del componente Presentación



Fuente: Autor.

**P2.** ¿Las imágenes contenidas en el taller facilitan la comprensión de los conceptos estudiados en el mismo?

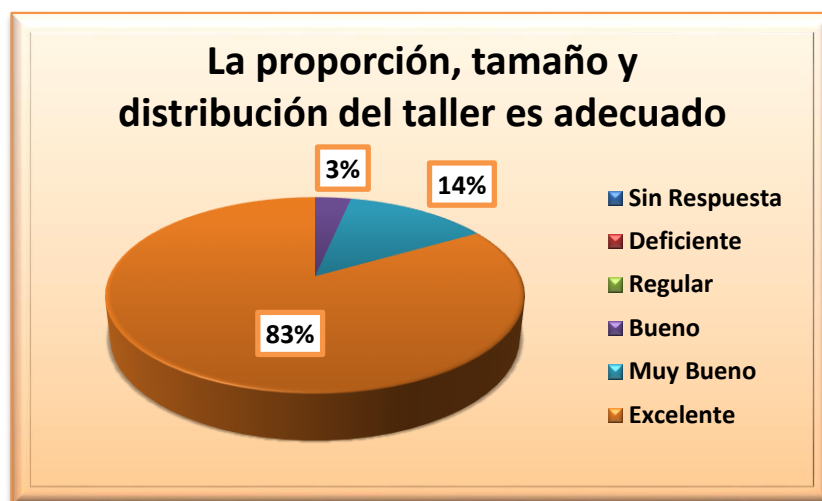
Figura 23. Resultados pregunta 2 del componente Presentación



Fuente: Autor.

**P3.** ¿La proporción, tamaño y distribución del taller es adecuado?

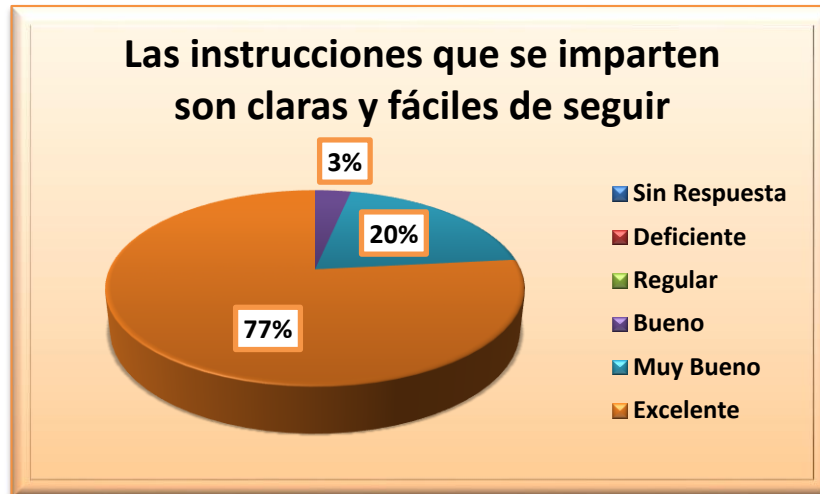
Figura 24. Resultados pregunta 3 del componente Presentación



Fuente: Autor.

**P4.** ¿Las instrucciones que se imparten son claras y fáciles de seguir?

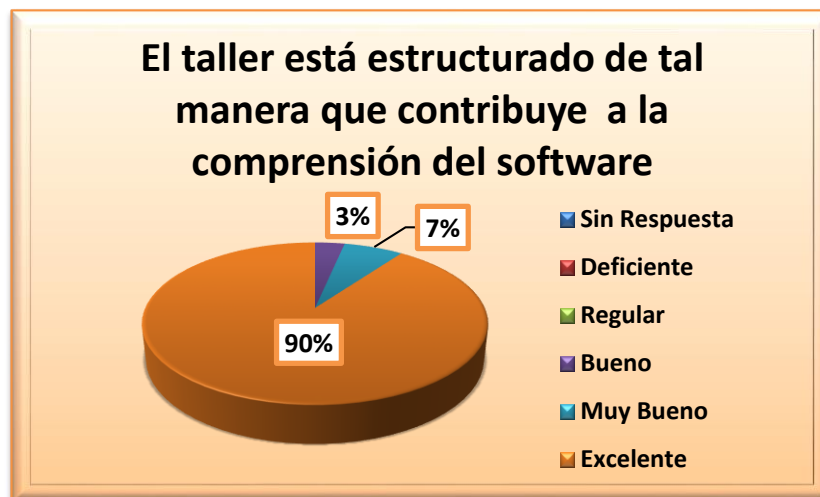
Figura 25. Resultados pregunta 4 del componente Presentación



Fuente: Autor.

**P5.** ¿El taller está estructurado de tal manera que contribuye a la comprensión del software?

Figura 26. Resultados pregunta 5 del componente Presentación



Fuente: Autor.

Resultados componente Presentación:

Tabla 14. Resultado general del componente Presentación

AFIRMACIÓN	PRESENTACIÓN						
	P1	P2	P3	P4	P5	TOTAL	%
<b>GRADO DE SATISFACIÓN</b>							
Sin Respuesta	0	0	0	0	0	0	
Deficiente	0	0	0	0	0	0	
Regular	0	0	0	0	0	0	
Bueno	0	0	1	1	1	3	2%
Muy Bueno	2	4	4	6	2	18	12%
Excelente	28	26	25	23	27	129	86%
<b>TOTAL</b>						<b>150</b>	

Fuente: Autor.

### 7.3.2 Lenguaje

P1. ¿El taller es explícito y contiene los elementos necesarios para realizar la práctica?

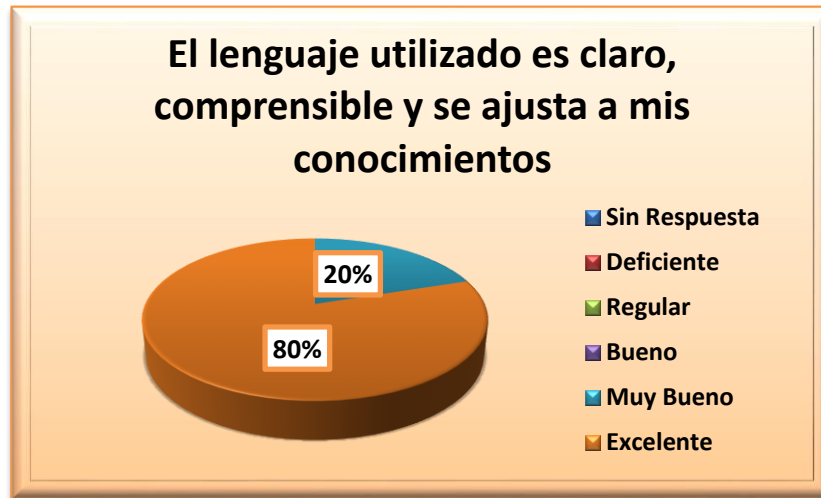
Figura 27. Resultados pregunta 1 del componente Lenguaje



Fuente: Autor.

**P2.** ¿El lenguaje utilizado es claro, comprensible y se ajusta a mis conocimientos?

Figura 28. Resultados pregunta 2 del componente Lenguaje



Fuente: Autor.

**P3.** ¿El taller es una herramienta útil que facilita el proceso de enseñanza-aprendizaje del software?

Figura 29. Resultados pregunta 3 del componente Lenguaje



Fuente: Autor.

Resultados componente Lenguaje:

Tabla 15. Resultado general del componente Lenguaje

AFIRMACIÓN	LENGUAJE				
	P1	P2	P3	TOTAL	%
<b>GRADO DE SATISFACIÓN</b>					
Sin Respuesta	0	0	0	0	
Deficiente	0	0	0	0	
Regular	0	0	0	0	
Bueno	0	0	1	1	1%
Muy Bueno	2	4	4	10	11%
Excelente	28	26	25	79	88%
<b>TOTAL</b>				<b>90</b>	

Fuente: Autor.

### 7.3.3 Cobertura y Profundidad

P1. ¿Los temas abordados son relevantes para el desarrollo de los talleres?

Figura 30. Resultados pregunta 1 del componente Cobertura y Profundidad



Fuente: Autor.

**P2.** ¿El tema presentado se explica de manera detallada?

Figura 31. Resultados pregunta 2 del componente Cobertura y Profundidad



Fuente: Autor.

**P3.** ¿Los ejemplos expuestos son claros y útiles?

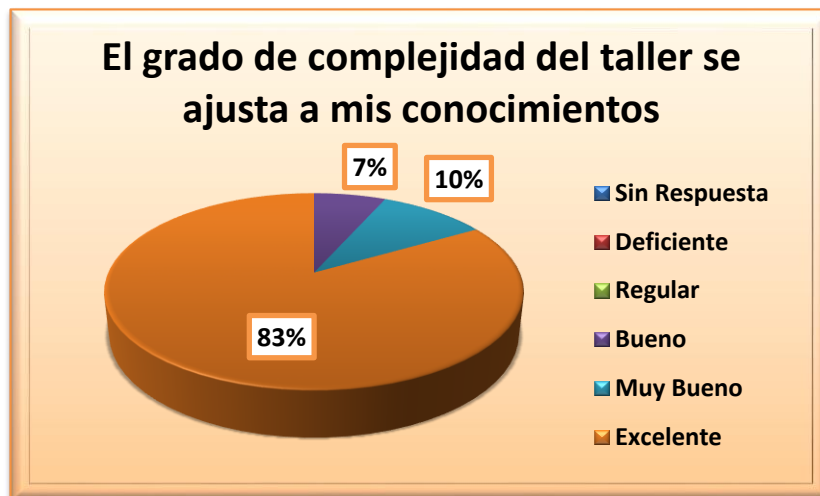
Figura 32. Resultados pregunta 3 del componente Cobertura y Profundidad



Fuente: Autor.

**P4.** ¿El grado de complejidad del taller se ajusta a mis conocimientos?

Figura 33. Resultados pregunta 4 del componente Cobertura y Profundidad



Fuente: Autor.

Resultados componente Cobertura y Profundidad:

Tabla 16. Resultado general del componente Cobertura y Profundidad

AFIRMACIÓN	COBERTURA Y PROFUNDIDAD					
	P1	P2	P3	P4	TOTAL	%
<b>GRADO DE SATISFACIÓN</b>						
Sin Respuesta	0	0	0	0	0	
Deficiente	0	0	0	0	0	
Regular	0	0	0	0	0	
Bueno	0	1	0	2	3	3%
Muy Bueno	3	2	4	3	12	10%
Excelente	27	27	26	25	105	88%
<b>TOTAL</b>					<b>120</b>	

Fuente: Autor.

### 7.3.4 Utilidad

**P1.** ¿La interfaz que maneja el software es amigable y facilita el proceso enseñanza-aprendizaje?

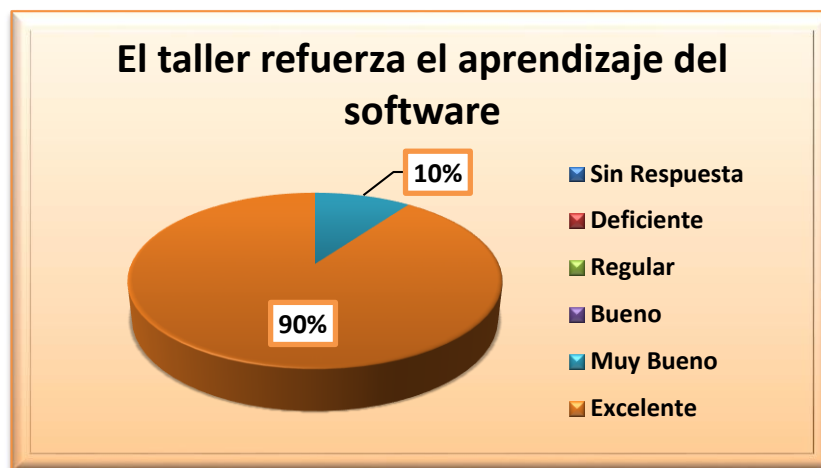
Figura 34. Resultados pregunta 1 del componente Utilidad



Fuente: Autor.

**P2.** ¿El taller refuerza el aprendizaje del software?

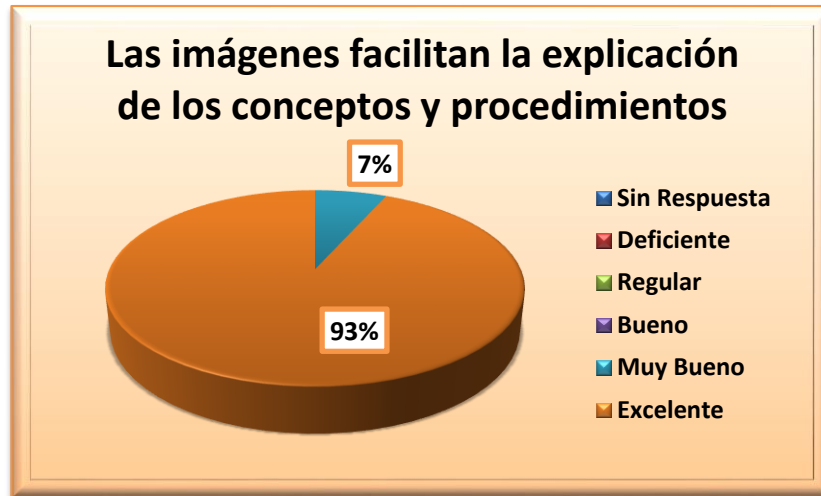
Figura 35. Resultados pregunta 2 del componente Utilidad



Fuente: Autor.

**P3.** ¿Las imágenes facilitan la explicación de los conceptos y procedimientos?

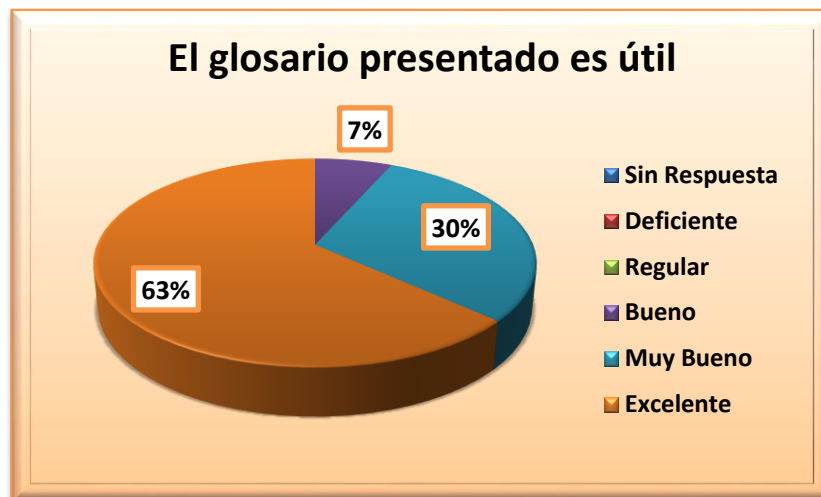
Figura 36. Resultados pregunta 3 del componente Utilidad



Fuente: Autor.

**P4.** ¿El glosario presentado es útil?

Figura 37. Resultados pregunta 4 del componente Utilidad



Fuente: Autor.

**P5.** ¿El software utilizado en el taller es una buena herramienta para enseñar Simulación?

Figura 38. Resultados pregunta 5 del componente Utilidad



Fuente: Autor.

Resultados componente Utilidad:

Tabla 17. Resultado general del componente Utilidad

AFIRMACIÓN	UTILIDAD					TOTAL	%
	P1	P2	P3	P4	P5		
<b>GRADO DE SATISFACIÓN</b>							
Sin Respuesta	0	0	0	0	0	0	
Deficiente	0	0	0	0	0	0	
Regular	0	0	0	0	0	0	
Bueno	1	0	0	2	1	4	3%
Muy Bueno	1	3	2	9	3	18	12%
Excelente	28	27	28	19	26	128	85%
<b>TOTAL</b>						<b>150</b>	

Fuente: Autor.

## 7.4 CONCLUSIONES DE LA EVALUACIÓN DE LA PRUEBA PILOTO

- ❖ Los estudiantes calificaron de excelente o muy bueno el componente presentación de los talleres con un 86% y 12% respectivamente, el 2% restante lo calificaron como bueno. Este componente abarca la presentación general, proporción, distribución de las figuras e instrucciones de los talleres diseñados.
- ❖ Los estudiantes calificaron de excelente o muy bueno el componente lenguaje de los talleres con un 88% y 11% respectivamente, solo el 1% restante lo calificaron como bueno. Este componente evalúa el lenguaje empleado en los talleres y si es comprensible y explicito, con lo cual se resalta el énfasis en la claridad de las instrucciones impartidas.
- ❖ El 88% de los estudiantes calificaron de excelente el componente cobertura y profundidad, un 10% como muy bueno y el 3% restante como bueno. Este componente evalúa los temas abordados y su explicación, ejemplos impartidos en la sala de cómputo donde realizan sus prácticas y el grado de complejidad de los mismos.
- ❖ El 85% de los estudiantes calificaron de excelente el componente utilidad, un 12% como muy bueno y el 3% restante como bueno. Este componente evalúa la utilidad del software Flexsim para aprender simulación y la aplicación de los talleres como herramienta de enseñanza.

## 8. CONCLUSIONES

- ❖ El instructivo y los talleres de este trabajo de grado han servido de apoyo a proyectos de grado exitosos de la Escuela de Estudios Industriales y Empresariales, donde fue necesaria la realización de simulaciones para representar sistemas reales complejos en el software Flexsim y tomar las mejores decisiones sin implementar cambios en el sistema de estudio.
- ❖ Se pudo establecer que la enseñanza de Flexsim es una propuesta que se percibe importante en el proceso de formación, no sólo porque que los estudiantes podrán utilizar software de última generación, sino complementar sus conocimientos teóricos y contrastarlos con la práctica; además les permitirá familiarizarse con los ambientes reales de manufactura en el uso de los equipos, análisis de datos (de entrada y de salida) y procesos.
- ❖ Las TICS proporcionan herramientas muy útiles para la enseñanza de software y la solución de problemas de la ingeniería industrial. Este trabajo de grado aporta una instrucción práctica de una potente herramienta para simulación de sistemas productivos, como es el software especializado en simulación FLEXSIM, mediante la utilización de software de gestión de aulas que facilita el aprendizaje colaborativo (Faronics Insight), software LMS para gestionar los contenidos (Plataforma Moodle) y aplicaciones informáticas para la elaboración de contenido multimedia como video tutoriales.
- ❖ El instructivo de aprendizaje utilizado como la estrategia principal en la enseñanza del software Flexsim contribuyó al desarrollo de los talleres brindando una guía y una base teórica para su adecuada implementación.

- ❖ Con la aplicación de la evaluación de los talleres cabe indicar la gran aceptación que tuvo los talleres piloto entre los estudiantes, pues la competencia No. 5 que se refiere a la evaluación de la “UTILIDAD” que los estudiantes perciben de los talleres en general, fue de un nivel de satisfacción “EXCELENTE” correspondiente al 85%.
  
- ❖ Este trabajo de grado sirve para capacitar a estudiantes de pregrado y de la maestría de Ingeniería Industrial que requieran de una herramienta metodológica para el aprendizaje de este software de simulación.

## 9. RECOMENDACIONES

- ❖ Es importante que los estudiantes refuercen sus conocimientos de programación de computadores o que no tomen el respectivo curso en semestres inferiores, dado que no es tan necesario en esa instancia y así poder llegar con más fundamentos de programación, no solamente a los talleres de Flexsim, sino a las asignaturas de semestres superiores que involucren el aprendizaje de un software especializado de simulación, optimización, macros con Excel, entre otros, ya que estos necesitan fundamentos básicos de programación y de esta manera sacar un mejor provecho de estas aplicaciones informáticas.
- ❖ Se recomienda utilizar herramientas de simulación desde materias tempranas de la ingeniería industrial como investigación de operaciones, estadística y producción para que el estudiante tenga otro punto de vista de cómo resolver un problema y poder contrastar la teoría con sus modelos de simulación.
- ❖ Es necesario utilizar un servidor dedicado para la licencia de Flexsim, de tal manera que el equipo de cómputo del docente sea un equipo cliente de la licencia y no presente bloqueos o lags del software, haciendo más fluido el trabajo.
- ❖ Una de las dificultades presentes en todo momento con el proyecto y el aprendizaje de los estudiantes, es la utilización de la licencia de Flexsim por fuera de clases, porque sin tener el software licenciado no se pueden estudiar los talleres vistos en la sala de cómputo en su totalidad, acceder a algunos elementos básicos de un modelo de simulación y a los módulos importantes de la aplicación como ExpertFit, Experimenter, OptQuest, etc., por lo cual se recomienda asignar turnos de trabajo con Flexsim para que los estudiantes

puedan reforzar el aprendizaje de este software de manera extra clase, acceder a la licencia y realizar los trabajos de la asignatura que impliquen la utilización de Flexsim.

- ❖ Una alternativa mejor a la propuesta anterior es la utilización de una VPN (Virtual Private Network) o Red Privada Virtual, donde se asignen grupos de trabajo de 2 o máximo 3 estudiantes para que accedan con un solo computador a la red VPN que enlace o extienda la red LAN de la sala de cómputo al computador registrado de cada grupo. Se asignaría un horario de trabajo distinto al de clases para que accedan remotamente a la licencia. El beneficio de esta recomendación es que puedan trabajar el día sábado mientras el servidor dedicado esté encendido y la sala operando, además la red virtual se puede desconectar (por fuera del horario establecido) para que no interfiera con las clases y los usuarios de Flexsim en la EEIE, sean estos estudiantes de otros niveles, personal administrativo y docentes. Posteriormente cuando finalice el semestre o la asignatura se eliminan o expulsan estos usuarios de esta red virtual, logrando admitir nuevos miembros cada semestre. La anualidad de esta red virtual es muy económica, siendo una alternativa viable.
  
- ❖ La utilización de una VPN no solo beneficiaría a los estudiantes de TMO, sino que podría aplicarse a otras asignaturas que involucren software licenciado (de tipo concurrente) de la EEIE, como Preactor y Minitab.

## BIBLIOGRAFÍA

- GARCÍA, Francisco; SIERRA, Jorge y GUZMÁN Virginia. Simulación de Sistemas para Administración e Ingeniería. CECSA, México, 2005.
- GUASH, Antoni; PIERA, Miguel Ángel; CASANOVAS, Josep y FIGUERAS, Jaume. Modelado y simulación. Aplicación y procesos logísticos de fabricación y servicios. Alfaomega, México, 2005.
- GARCÍA, Eduardo; GARCÍA, Heriberto y CARDENAS, Leopoldo. Simulación y análisis de sistemas con ProModel. Pearson Educación, México, 2006.
- BLANCO Rivero, Luis; FAJARDO Piedrahita, Iván. Simulación con ProModel: casos de producción y logística. Escuela Colombiana de Ingeniería, Bogotá, 2003.
- RÍOS INSÚA, David; RÍOS INSÚA, Sixto; MARTÍN JIMÉNEZ, Jacinto y JIMÉNEZ MARTÍN, Antonio. Simulación Métodos y Aplicaciones. Segunda Edición, Editorial Alfaomega, Madrid, 2009.
- GARAVITO HERNÁNDEZ, Edwin Alberto; ARENAS DÍAZ, Piedad. Simulación de Procesos de Manufactura 1. Publicaciones UIS, Bucaramanga, 2008.
- HARREL, Charles; GHOSH Biman. Simulation using ProModel. First Edition, McGraw Hill, United States of America, 2006.
- KELTON, W. David; SADOWSKI, Randall and SADOWSKI, Deborah. Simulation with Arena. Second Edition, McGraw Hill, United States of America, 2000.

- LAW, Averill; KELTON, W. David. Simulation Modeling and Analysis. Third Edition, McGraw Hill, United States of America, 2000.
- BANKS, Jerry; CARSON II, John S.; NELSON, Barry L. and NICOL, David M. Discrete-Event System Simulation. Third Edition, Prentice Hall, United States of America, 2001.
- ROSS, Sheldon M. Simulación, Segunda Edición, Prentice Hall, México, 1999.
- CORREDOR, Martha Vitalia; PEREZ, Martha Ilce, ARBELAEZ, Ruby. Estrategias de enseñanza y aprendizaje. CEDEDUIS, Bucaramanga, 2009.
- FABRA, Maria. Técnicas de grupo para la cooperación. Ediciones CEAC, Barcelona, 1994.
- 2012 Winter Simulation Conference, página oficial de la conferencia anual de simulación. Consultado en Julio de 2012. Disponible en:  
<http://www.wintersim.org>
- Flexsim | Vatic Consulting Group, página oficial de la empresa proveedora de Flexsim en Colombia. Consultado en Julio de 2012. Disponible en:  
<http://www.vaticgroup.com/unlimitpages.asp?id=19>
- Flexsim Simulation Software, página oficial del software. Consultado en Febrero de 2012. Disponible en:  
<http://www.flexsim.com/>

- Flexsim South America, página oficial de Flexsim en Suramérica. Consultado en Febrero de 2012. Disponible en:  
<http://www.flexsim.cl/>
- Flexsim Entrenamiento Online, página de entrenamiento en Flexsim, en español. Consultado en Febrero de 2012. Disponible en:  
<http://www.competition.flexsim.com/entrenamiento/>
- Página oficial de ProModel. Consultado en Marzo de 2012. Disponible en:  
<http://www.promodel.com/>
- Arena Simulation Software, página oficial del simulador Rockwell Arena. Consultado en Marzo de 2012. Disponible en:  
<http://www.arenasimulation.com>
- Process Simulation Modeling Software, página oficial de Witness. Consultado en Marzo de 2012. Disponible en:  
<http://www.lanner.com/en/media/witness/witness-pwe.cfm>
- Página oficial de AutoMod. Consultado en Marzo de 2012. Disponible en:  
<http://www.appliedmaterials.com/services-software/automation-software>
- ❖ SOFTWARE Shop / El Distribuidor de software científico líder en Latinoamérica. Consultado en Marzo de 2012. Disponible en:  
<http://www.software-shop.com>
- SNIES – Sistema Nacional de Información de la Educación Superior. Consultado en Junio de 2012. Disponible en:  
<http://snies.mineducacion.gov.co/ConsultaSnies/ConsultaSnies/consultarInfoProgramasAcademicos.jsp>

**ANEXO A**  
**Instructivo de Aprendizaje del Software de Simulación Flexsim**



TÉCNICAS MODERNAS DE  
OPTIMIZACIÓN

2013

INSTRUCTIVO DE APRENDIZAJE  
DEL SOFTWARE DE SIMULACIÓN

***Flexsim***

WILLIAM JESÚS PINTO TRIANA  
Ingeniero Industrial  
EEIE - UIS

**INSTRUCTIVO DE APRENDIZAJE DEL  
SOFTWARE DE SIMULACIÓN FLEXSIM**



**WILLIAM JESUS PINTO TRIANA**

**DOCENTE**

**ING. EDWIN ALBERTO GARAVITO HERNÁNDEZ**



**UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS  
ESCUELA DE ESTUDIOS INDUSTRIALES Y EMPRESARIALES  
BUCARAMANGA**

**2013**

## TABLA DE CONTENIDO

	pág.
<b>INTRODUCCIÓN.....</b>	<b>12</b>
<b>1. ACERCA DE FLEXSIM .....</b>	<b>13</b>
1.1 APLICACIONES.....	14
1.2 MÓDULOS .....	16
1.3 COMPONENTES.....	17
<b>2. INTERFAZ GRÁFICA DE FLEXSIM .....</b>	<b>24</b>
2.1 BARRA DE MENÚ.....	29
2.2 BARRA DE HERRAMIENTAS .....	44
2.3 PANEL DE EJECUCIÓN DE LA SIMULACIÓN .....	46
2.4 LIBRERÍA DE OBJETOS .....	47
2.4.1 Árbol de la Familia Flexsim .....	48
2.4.2 Objetos Discretos.....	49
2.5 VISTAS DEL MODELO.....	56
<b>3. CONSTRUCCIÓN DE MODELOS .....</b>	<b>59</b>
3.1 ELEMENTOS DE UN MODELO DE SISTEMA .....	59
3.2 TERMINOLOGÍA DEL SOFTWARE FLEXSIM.....	61
3.3 NAVEGACIÓN EN LA VISTA DE MODELO.....	66
3.4 CREACIÓN DE OBJETOS.....	71
3.5 EDICIÓN DE OBJETOS .....	74
3.6 VENTANA DE PROPIEDADES .....	79
3.7 LISTA DE ESTADOS DE LOS OBJETOS.....	85
3.8 CONEXIÓN DE PUERTOS .....	92
3.9 MOVIMIENTO DE FLOWITEMS .....	94



<b>4. ADICIONANDO LA LÓGICA DEL MODELO .....</b>	<b>97</b>
4.1 CONFIGURACIÓN DE LLEGADAS Y TIEMPOS DE OPERACIÓN .....	97
4.2 ASIGNANDO ATRIBUTOS A OBJETOS .....	102
4.3 CREANDO VARIABLES GLOBALES .....	103
4.4 EVENTOS DISPARADORES O TRIGGERS.....	105
4.5 FLUJO DEL MODELO.....	110
4.6 TABLAS GLOBALES .....	112
4.7 AGRUPAR Y DESAGRUPAR FLOWITEMS .....	114
4.8 UTILIZAR TASK EXECUTERS .....	116
4.9 AGREGAR REDES DE TRANSPORTE .....	118
4.10 VISUALIZAR INFORMACIÓN EN PANTALLA .....	119
4.11 DASHBOARD.....	120
4.11.1 Tipos de Gráficos .....	121
<b>5. UTILIZACIÓN DE LAS HERRAMIENTAS DE FLEXSIM .....</b>	<b>129</b>
5.1 EXPERTFIT .....	129
5.2 EXPERIMENTER .....	130
<b>BIBLIOGRAFÍA .....</b>	<b>131</b>

## LISTADO DE TABLAS

Tabla 1. Aplicaciones de Servicios .....	14
Tabla 2. Aplicaciones de Manufactura .....	15
Tabla 3. Aplicaciones de Logística y Manejo de Materiales.....	15
Tabla 4. Pestañas compartidas de propiedades.....	80
Tabla 5. Pestañas únicas de propiedades .....	81
Tabla 6. Lista de estados de los objetos en Flexsim.....	87

## LISTADO DE FIGURAS

Figura 1. Modelos en Flexsim .....	19
Figura 2. Entorno y objetos gráficos de modelos en Flexsim.....	20
Figura 3. Aplicación en logística y salud .....	21
Figura 4. Warehousing o Almacenamiento .....	22
Figura 5. Ejemplo de Modelo de Simulación en Flexsim .....	23
Figura 6. Ventana inicial de Flexsim .....	24
Figura 7. Unidades del modelo .....	25
Figura 8. Actualizar modelo .....	25
Figura 9. Manual de usuario de Flexsim .....	26
Figura 10. Ventana principal de Flexsim.....	27
Figura 11. Interfaz Gráfica de Flexsim .....	28
Figura 12. Barra de menú .....	29
Figura 13. Menú File   Advanced Files.....	29
Figura 14. Model Settings .....	32
Figura 15. Ajuste de Number Precision y Conveyor Angle .....	33
Figura 16. Menú Edit.....	34
Figura 17. Menú View .....	34
Figura 18. Utilidades del submenú Modeling Utilities.....	37
Figura 19. Utilidades Tree Navigation y Travel Networks .....	38
Figura 20. Utilidades View y Model Layouts .....	39
Figura 21. Menú Build.....	39
Figura 22. Menú Execute .....	40
Figura 23. Menú Debug .....	40
Figura 24. Menú Window .....	41
Figura 25. Menú Help .....	41
Figura 26. Menú Statistics .....	42
Figura 27. Menú Tools.....	43

Figura 28. Barra de herramientas (toolbar) .....	44
Figura 29. Texto descriptivo de la toolbar .....	44
Figura 30. Personalización de la barra de herramientas (toolbar) .....	45
Figura 31. Barra de herramientas personalizada .....	45
Figura 32. Panel de ejecución de la simulación .....	46
Figura 33. Librería de Objetos Discretos y de Fluidos .....	47
Figura 34. Árbol de la familia Flexsim para objetos discretos .....	48
Figura 35. Vista 3D del modelo (vista frontal o front) .....	56
Figura 36. Vista 3D del modelo (superior o top).....	57
Figura 37. Vista Planar del modelo (diagrama en bloques) .....	57
Figura 38. Model Views .....	58
Figura 39. Utilidad Views .....	58
Figura 40. Objetos de Flexsim .....	61
Figura 41. Contenedor de Flowitems .....	63
Figura 42. Ítems de Flujo o Flowitems .....	63
Figura 43. Clases de Objetos Discretos.....	64
Figura 44. Puertos en Flexsim .....	65
Figura 45. Vista inicial del modelo .....	66
Figura 46. Desplazamiento del layout en el plano X-Y .....	67
Figura 47. Rotación del layout en X-Y-Z.....	68
Figura 48. Botón scroll del mouse.....	69
Figura 49. Zoom de acercamiento .....	69
Figura 50. Modo de vuelo .....	70
Figura 51. Reinicio de la vista de modelo .....	71
Figura 52. Creación de Objetos – Paso 1: Clic .....	72
Figura 53. Creación de Objetos – Paso 2: Arrastrar .....	72
Figura 54. Creación de Objetos – Paso 3: Soltar.....	73
Figura 55. Creando objetos .....	73
Figura 56. Renombrando objetos.....	74
Figura 57. Seleccionando objetos.....	75

Figura 58. Moviendo objetos.....	76
Figura 59. Desplazamiento en el eje Z .....	77
Figura 60. Rotación de objetos .....	78
Figura 61. Rotación de objetos .....	78
Figura 62. Propiedades del objeto .....	79
Figura 63. Pestaña General.....	82
Figura 64. Pestaña Labels .....	83
Figura 65. Pestaña Statistics .....	84
Figura 66. Pestaña Flow .....	84
Figura 67. Pestaña Breakdowns .....	85
Figura 68. Principales estados de los objetos discretos en Flexsim .....	86
Figura 69. Teclas para conexión y desconexión de puertos .....	93
Figura 70. Conexión y desconexión de puertos .....	93
Figura 71. Pestaña Conveyor .....	95
Figura 72. Movimiento de Flowitems .....	96
Figura 73. Tipos de llegada .....	97
Figura 74. Tiempo entre llegadas .....	98
Figura 75. Asignación del tiempo entre arribos.....	99
Figura 76. Tiempo de operación .....	100
Figura 77. Asignación del tiempo de operación .....	101
Figura 78. Pestaña Labels de un Flowitem.....	103
Figura 79. Creación de variables globales.....	104
Figura 80. Tipos de datos .....	104
Figura 81. Ejemplo 1, Paso 4.....	106
Figura 82. Selección de distribución estadística .....	107
Figura 83. Sintaxis de la distribución exponencial .....	107
Figura 84. Lista desplegable en el trigger OnCreation.....	108
Figura 85. Trigger Set Item Type and Color.....	108
Figura 86. Itemtype del flowitem Box .....	109
Figura 87. Pestaña de flujo de los objetos .....	110

Figura 88. Tablas Globales.....	112
Figura 89. Asignando datos de texto en Tablas Globales.....	113
Figura 90. Ejemplo de Tabla Global.....	113
Figura 91. Ejemplo de Combiner .....	114
Figura 92. Modo de combinar flowitems .....	115
Figura 93. Separator .....	116
Figura 94. TaskExecutor .....	116
Figura 95. TaskExecutor .....	117
Figura 96. Utilización de operadores .....	117
Figura 97. Redes de transporte .....	118
Figura 98. Tipo de conexión entre nodos.....	119
Figura 99. Visual Display .....	119
Figura 100. Dashboard .....	120
Figura 101. Dashboard – Text Panels .....	122
Figura 102. Dashboard – Bar Graphs.....	123
Figura 103. Dashboard – Pie Charts.....	124
Figura 104. Dashboard – Line Graphs.....	124
Figura 105. Listas desplegadas o Picklist .....	126
Figura 106. Ventanas emergentes o popups .....	126
Figura 107. Ventana de edición de código.....	128

## GLOSARIO

- ❖ **ADD-ON:** llamado también agregado o plugin, es una extensión o programa que solo funciona anexo a otra y que sirve para incrementar o complementar sus funcionalidades. Tiene funciones más extensas que el plugin. Abarca plugins, extensiones, temas para las aplicaciones diseñadas.
  
- ❖ **APLICACIÓN INFORMÁTICA:** es un tipo de programa informático diseñado como herramienta para permitir a un usuario realizar uno o diversos tipos de trabajos.
  
- ❖ **BARRA DE HERRAMIENTAS:** conocida en inglés como *toolbar*, es un componente de una interfaz gráfica de usuario mostrada usualmente en pantalla a modo de fila, columna, o bloque que contiene iconos o botones que, al ser presionados, activan ciertas funciones de una aplicación.
  
- ❖ **BARRA DE MENÚ:** es un área de la interfaz de usuario que indica y presenta las opciones o herramientas de una aplicación informática, dispuestas en menús desplegables.
  
- ❖ **DRAG AND DROP:** arrastrar y soltar. Es una expresión informática que se refiere a la acción de mover con el ratón objetos de una ventana a otra o entre partes de una misma ventana. La secuencia básica involucrada en arrastrar-y-soltar es:
  - ✓ Presionar, y mantener, el botón del mouse u otro dispositivo apuntador, para "tomar" el objeto,
  - ✓ "Arrastrar" el objeto/cursor/dispositivo apuntador a la ubicación deseada,
  - ✓ "Soltar" el objeto soltando el botón.

- ❖ **ENTORNO DE DESARROLLO INTEGRADO (IDE):** conocida también como *IDE* (del inglés Integrated Development Environment) es una aplicación informática que proporciona un conjunto de herramientas de programación para el desarrollo de software. Normalmente se compone de: un editor de código fuente, un compilador y/o intérprete, un depurador y un constructor de interfaz gráfica de usuario (GUI).
  
- ❖ **INTERFAZ DE USUARIO:** es el medio con que el usuario puede comunicarse con una máquina, un equipo o un computador, y comprende todos los puntos de contacto entre el usuario y el equipo. Normalmente suelen ser fáciles de entender y fáciles de accionar. Las interfaces básicas de usuario son aquellas que incluyen elementos como menús, ventanas, teclado, ratón, los beeps y algunos otros sonidos que la computadora hace, y en general, todos aquellos canales por los cuales se permite la comunicación entre el ser humano y la computadora.
  
- ❖ **INTERFAZ GRÁFICA DE USUARIO (GUI):** conocida también como *GUI* (del inglés Graphical User Interface) es un programa informático que actúa de interfaz de usuario, utilizando un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz. Su principal uso, consiste en proporcionar un entorno visual sencillo para permitir la comunicación con el sistema operativo de una máquina o computador.
  
- ❖ **LAYOUT:** esquema de distribución de los elementos dentro un diseño.
  
- ❖ **MENÚ:** es una serie de opciones que el usuario puede elegir para realizar determinada tarea.
  
- ❖ **MENÚ CONTEXTUAL:** menú emergente de contexto que aparece cuando el usuario presiona el botón derecho del mouse o hace clic derecho.

- ❖ **MODELO:** es una representación de un sistema con el fin de estudiarlo.
- ❖ **MODELO CONCEPTUAL:** proporciona un marco de referencia, estructurado y claramente definido del modelo a simular.
- ❖ **PLUG-IN:** llamado también complemento o plugin, es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica. Es un módulo de un programa.
- ❖ **SCROLL:** rueda de desplazamiento del mouse.
- ❖ **SISTEMA:** conjunto de elementos que se interrelacionan para funcionar como un todo con unas fronteras claras y definidas para el logro de un objetivo, por ejemplo: el sistema de atención de clientes en un banco, el sistema de inventarios de una empresa o el sistema de atención en la sala de emergencia de un hospital.
- ❖ **TMO:** Técnicas Modernas de Optimización. Electiva Técnica Profesional del plan de estudios de Ingeniería Industrial.

## INTRODUCCIÓN

El presente manual corresponde a una herramienta de apoyo para el desarrollo de la asignatura TÉCNICAS MODERNAS DE OPTIMIZACIÓN (TMO) de la Universidad Industrial de Santander (UIS).

El objetivo de la asignatura TMO es formar en el estudiante habilidades para analizar y construir modelos representativos de sistemas productivos y de servicios, reales o hipotéticos, generar la capacidad de diseñar modelos que involucren manejo de variables e información de entrada y salida a través de la simulación de eventos discretos.

Este instructivo de aprendizaje describe cómo utilizar las herramientas básicas del software de simulación Flexsim para simular sistemas discretos. Este material se complementa con el documento: “Talleres prácticos con el software de simulación Flexsim” como manual de prácticas. El material está disponible en la plataforma Moodle de la asignatura.

Por su diseño, este documento no reemplaza la extensa documentación de Flexsim, tales como su “*Guía de Usuario*” en inglés, donde encontrará una referencia a todos los comandos empelados por Flexsim, sino complementar en español la ayuda disponible y ser un instrumento de apoyo en la construcción de modelos de simulación.

Salvo que se indique lo contrario, las funciones descritas siempre se referirán a la versión 6.0.2 de Flexsim.

## 1. ACERCA DE FLEXSIM

El software de Simulación Flexsim es un programa de simulación de eventos discretos, orientado a objetos, que se utiliza para construir modelos que ayudan a visualizar flujos de procesos para optimizar throughput y minimizar gastos operativos.

Flexsim suministra a los usuarios una interfaz gráfica de usuario (GUI - *Graphical User Interface*) amena e intuitiva para visualizar, modelar y simular flujos de procesos usando el modo de tomar y arrastrar objetos (*Drag and Drop*) en un ambiente tridimensional. Incluye un análisis estadístico con detenimiento, de rendimientos de procesos, cuellos de botella y throughput.

Flexsim aporta una gran flexibilidad para simular cualquier tipo de proceso y la construcción de modelos de simulación con la mayor rapidez y con una facilidad de uso extraordinaria, sin necesidad de programación, aunque recibe código C/C++ y tiene un lenguaje propio llamado Flexscript, lo cual lo convierte en uno de los software de simulación más robustos en el mercado.

Permite evaluar diversos escenarios y responder a preguntas del tipo ¿qué pasa si...? de una manera rápida, precisa y lo más importante: libre de riesgos. Además la visualización de una simulación tridimensional animada es la manera más efectiva de explicar un nuevo concepto o proyecto. Un modelo de simulación en tres dimensiones impresiona y capta la atención de los asistentes, permite exponer tanto internamente como externamente un concepto o sistema propuesto y marcar la diferencia con la competencia.

Este simulador permite visualizar y probar cambios en las operaciones en cualquier proceso de manufactura, logística, manejo de materiales y de servicios, evitando los altos costos riesgos y extensos tiempos que conllevan el experimentar con cambios en el mundo real y su análisis por prueba y error, generando grandes ahorros. Cuenta con gráficos, reportes y estadísticas como resultados del modelo de simulación, además de importar datos de entrada desde cualquier programa o base de datos. También cuenta con la capacidad de crear videos de las simulaciones desde la misma aplicación que posteriormente se pueden presentar o compartir. Los objetos y el layout utilizados en el modelo pueden ser importados desde aplicaciones CAD, o utilizar la librería 3D que trae por defecto, tanto para modelos discretos como sistemas continuos.

## 1.1 APLICACIONES

Entre las aplicaciones de este software de simulación se encuentran:

**Tabla 1. Aplicaciones de Servicios**

SERVICIOS	
Inspecciones de seguridad en pasajeros y equipaje	Diseño de restaurantes de comida rápida
Mejoramiento de niveles de servicio	Hospitales y Centros de Emergencia
Sistemas militares	Centros de llamadas o Call Centers

Fuente: Flexsim.

**Tabla 1. Aplicaciones de Manufactura**

MANUFACTURA	
Lean Manufacturing	Automatización
Programación de la producción	Determinación de capacidades
Identificación de cuellos de botella	Mantenimientos
Diseño de layout	Reducción de tiempos de preparación
Balanceo de líneas	Análisis de Throughput
Determinación de capacidad	Planeación de personal y recursos
Líneas de ensamble	Determinación de turnos de operación
Reingeniería	Evaluación de Supply-Chain
Six-Sigma / Just-In-Time	Calidad
Robótica	Inspecciones y retrabajos

Fuente: Flexsim.

**Tabla 2. Aplicaciones de Logística y Manejo de Materiales**

LOGÍSTICA Y MANEJO DE MATERIALES	
Evaluación de sistemas de almacenamiento	Determinación de números de montacargas
Evaluación de sistemas de alistamiento	AGV o Vehículos Guiados Automáticamente
Evaluación de Supply Chain	Sistemas Robotizados ASRS
Tráfico y distribución	Determinación de puntos de entrega
Sistemas de transporte y bandas	Proceso de embarque
Elevadores	Justificación de Inversiones
Capacidad y tipos de Racks	Surtido de órdenes

Fuente: Flexsim.

## 1.2 MÓDULOS

### 1.2.1 ExpertFit

Es un módulo estadístico para realizar pruebas de bondad de ajuste, el cual determina cual es la distribución de probabilidad que mejor se ajusta a los datos de muestra, produciendo información muy importante para determinar las distribuciones asociadas a las variables aleatorias del modelo, como podría ser una distribución normal indicando los valores de la media y su desviación estándar o bien una exponencial por mencionar un ejemplo.

### 1.2.2 OptQuest

El módulo de optimización para los modelos de simulación automáticamente crea diferentes escenarios a partir de los modelos creados en Flexsim variando las distintas condiciones que se le especifiquen tales como el número de personas, montacargas, inventario, capacidad de las máquinas, velocidades de las bandas transportadoras, personal de mantenimiento, turnos, demandas o cualquier variable que se desee, utilizando algoritmos para evaluar esta gama de escenarios y encontrar automáticamente la solución óptima. Los criterios para encontrar esta solución óptima los define el usuario y pueden ser cualquiera de los objetivos que se deseen.

Ésta es una herramienta muy útil en el análisis posterior del modelo. Con ella se pueden diseñar experimentos destinados a conocer el impacto de factores críticos que se generan a partir de la variación de las variables aleatorias seleccionadas para ello. Asimismo, permite discernir cuál es la mejor combinación de factores para obtener el máximo beneficio al mejorar un proceso.

Ejemplos de los objetivos son maximizar utilidades, minimizar tiempos de surtido, minimizar los tiempos de espera, maximizar piezas producidas, maximizar el nivel de servicio o cualquier otro objetivo que se desee.

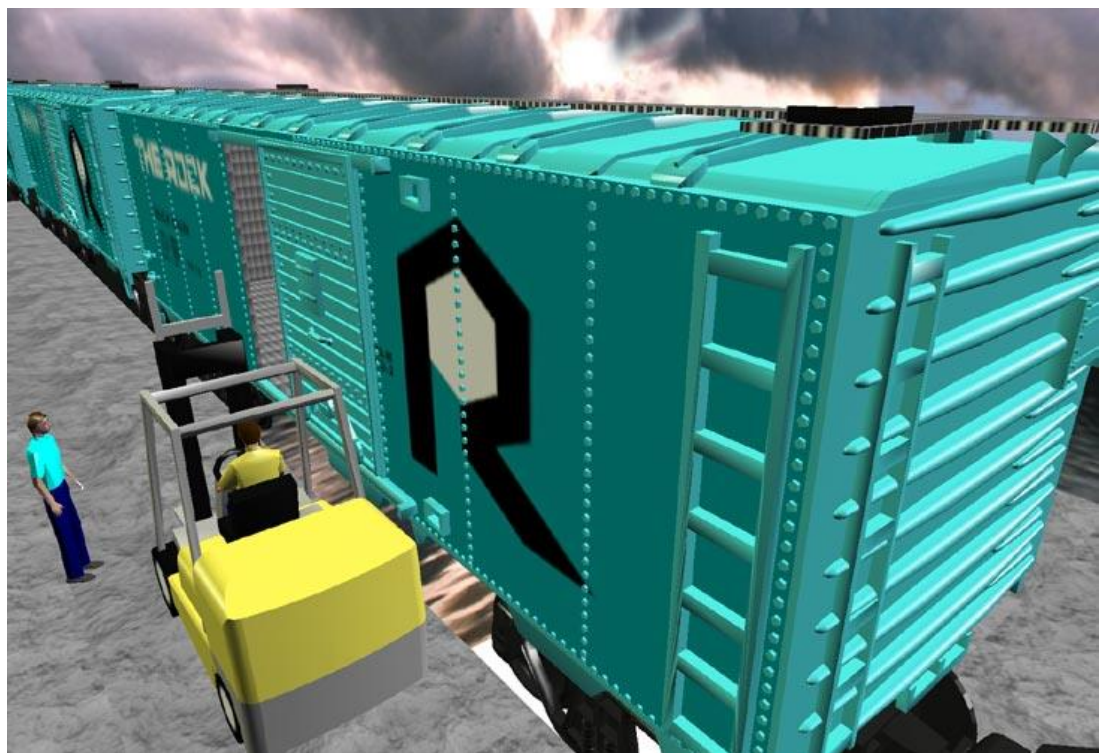
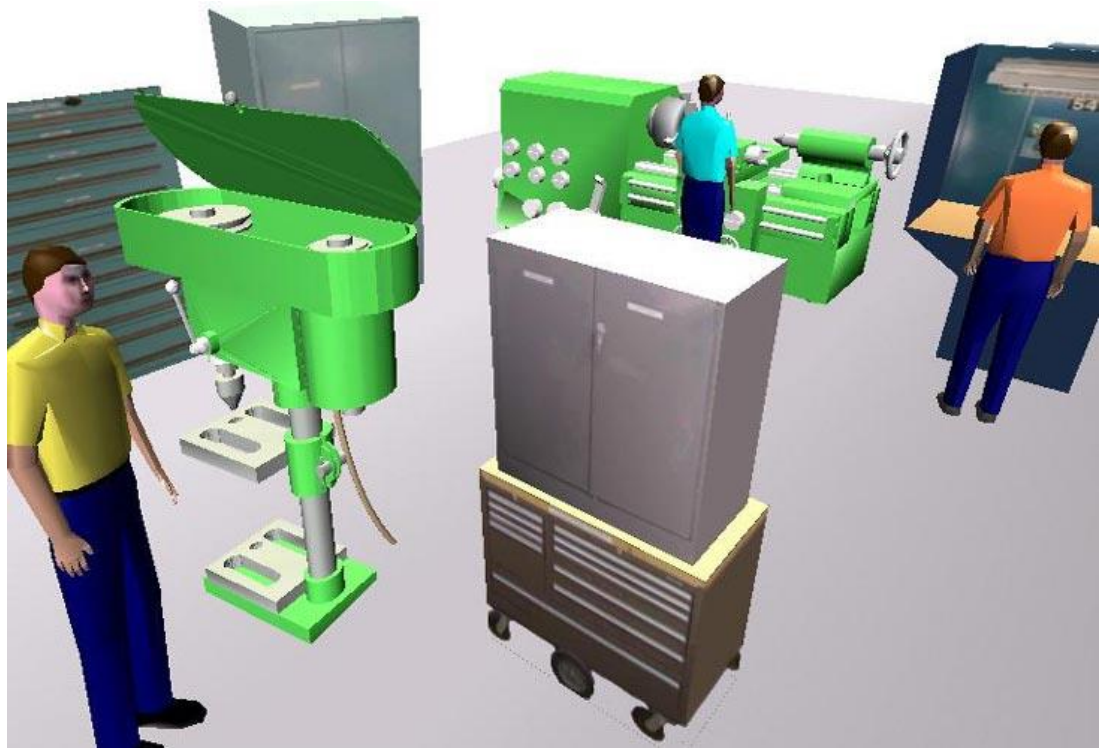
### 1.3 COMPONENTES

Entre los componentes de Flexsim encontramos:

- ❖ **Módulo de interface de la modelación - Flexsim.** Contiene un área de trabajo o layout donde se define el modelo y todos sus componentes. Los modelos se pueden visualizar en 3D y con vista planar o 2D.
- ❖ **Editor gráfico.** El editor gráfico de Flexsim cuenta con una serie de librerías en 3D que permiten dar una mejor representación visual a los modelos realizados. Flexsim cuenta con librerías para eventos discretos y otra para sistemas continuos. Se pueden crear librerías personalizadas, nuevos objetos, importar y exportarlos desde y hacia aplicaciones CAD.
- ❖ **Resultados.** Flexsim puede visualizar resultados en tiempo real, mostrando gráficas de estado de las variables y estadísticas para cada uno de los objetos definidos en el modelo. En este módulo se pueden ver los resultados de todas las variables del modelo. Además, el módulo permite la interacción con programas de hoja de cálculo, como Excel. Equivale al Procesador de salida e Interface de salida, puesto que permite al usuario visualizar de manera clara los resultados de la simulación a través de tablas y registros o de gráficos de estado.

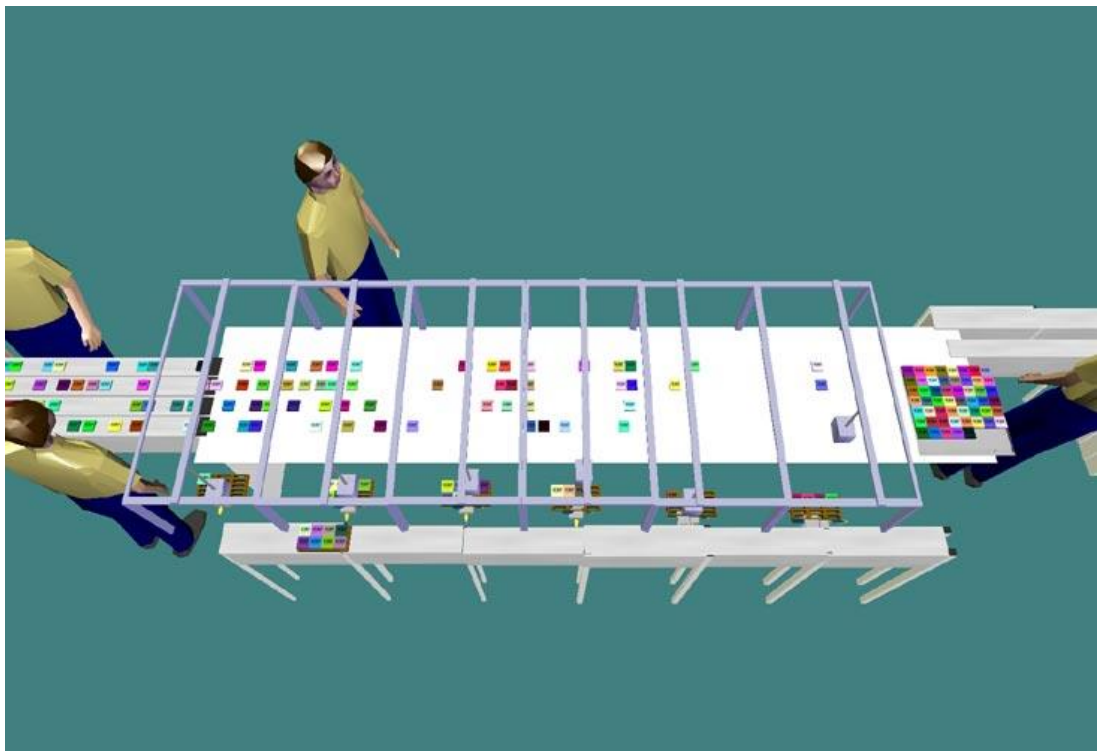
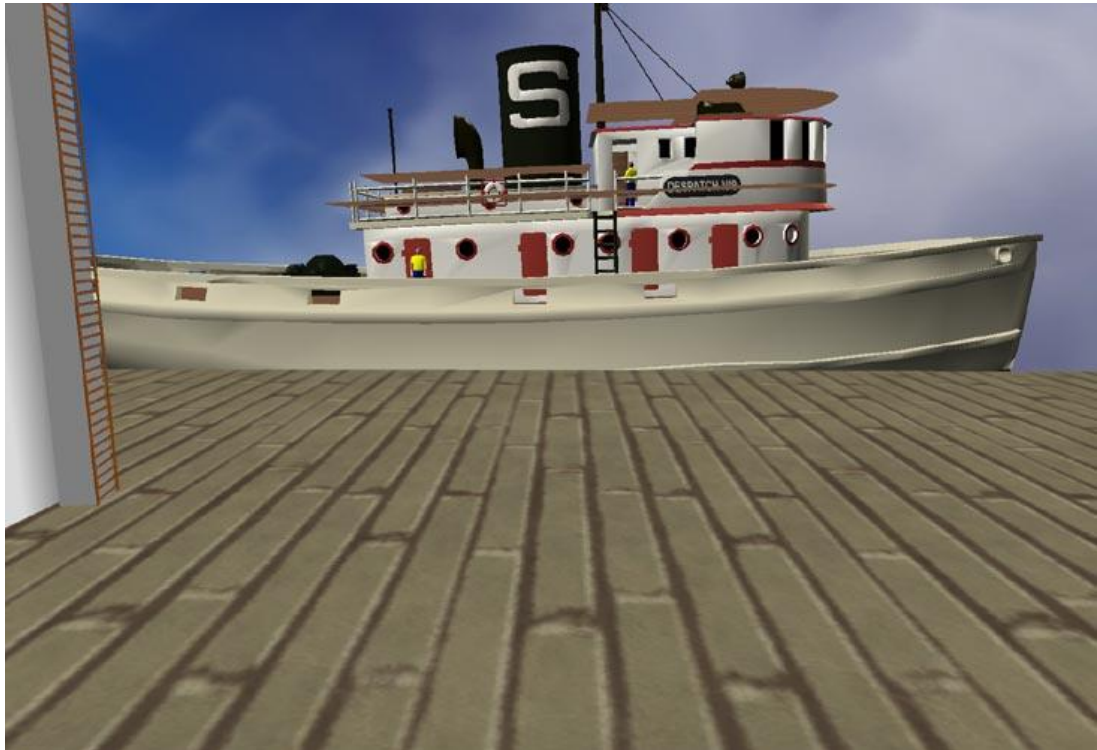
- ❖ **Referencias y ayuda.** Flexsim cuenta con un manual muy completo de ayuda, además puede consultarse también el portal online de este software de simulación.
- ❖ **Módulo de interface de la simulación.** Representa gráficamente la simulación que se visualiza durante la corrida de la simulación y que permite al usuario interactuar con esta para controlar efectos como su velocidad, visualizar el estado de las variables, entre otros. Tiene vistas en 3D y vista planar del modelo.
- ❖ **Procesador de animación.** Actualiza la representación gráfica de lo que se está simulando en el tiempo de reloj del sistema, de tal manera que se conoce que ocurre en cada momento de la simulación, a la vez que va generando los reportes estadísticos y resultados arrojados por el sistema simulado.
- ❖ **Procesador de la simulación.** Se compone de elementos como :
  - ✓ Reloj de la simulación: lleva el control del avance del tiempo. Los modelos pueden correrse o ejecutarse por horas, días, meses inclusive años.
  - ✓ Calendario de eventos: consiste en una lista que contiene los eventos que se ejecutarán en el orden cronológico en que esto sucederá. Esto se puede evidenciar cuando se ejecuta la simulación paso a paso.
  - ✓ Lógica del evento: contiene las instrucciones para simular el comportamiento de un evento.
  - ✓ Generador de números aleatorios: como todo software importante de simulación, Flexsim cuenta con algoritmos para generar series de números aleatorios a partir de diferentes semillas.

Figura 1. Modelos en Flexsim



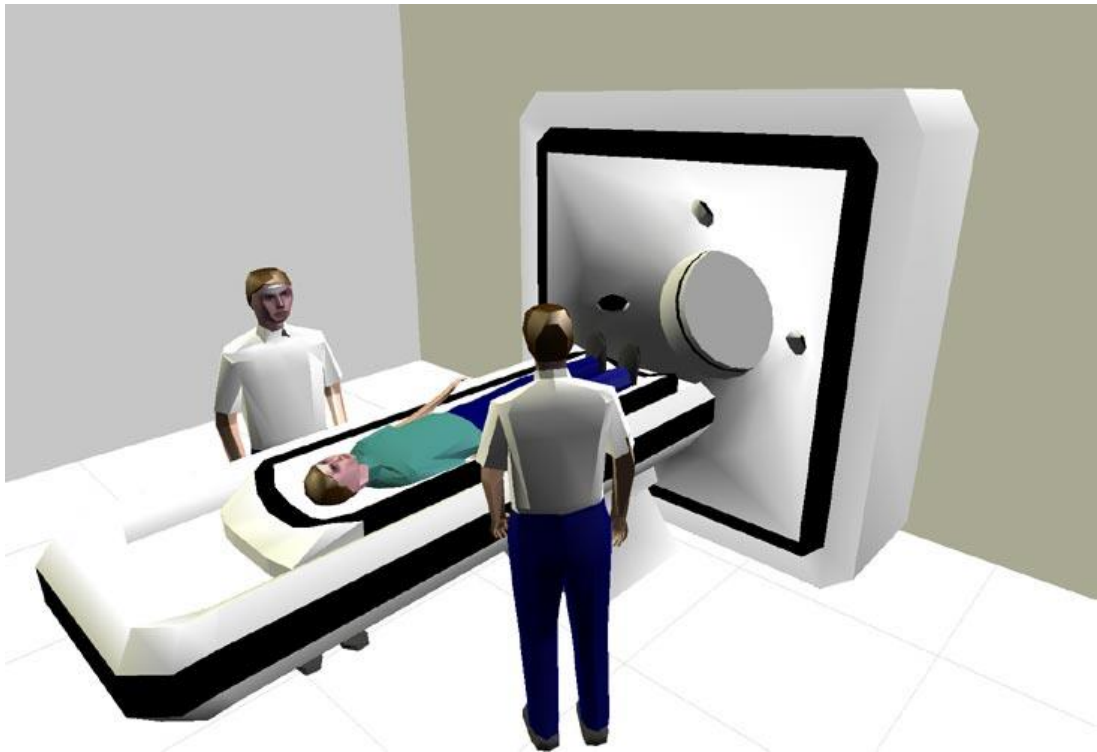
Fuente: Flexsim.

Figura 2. Entorno y objetos gráficos de modelos en Flexsim



Fuente: Flexsim.

Figura 3. Aplicación en logística y salud



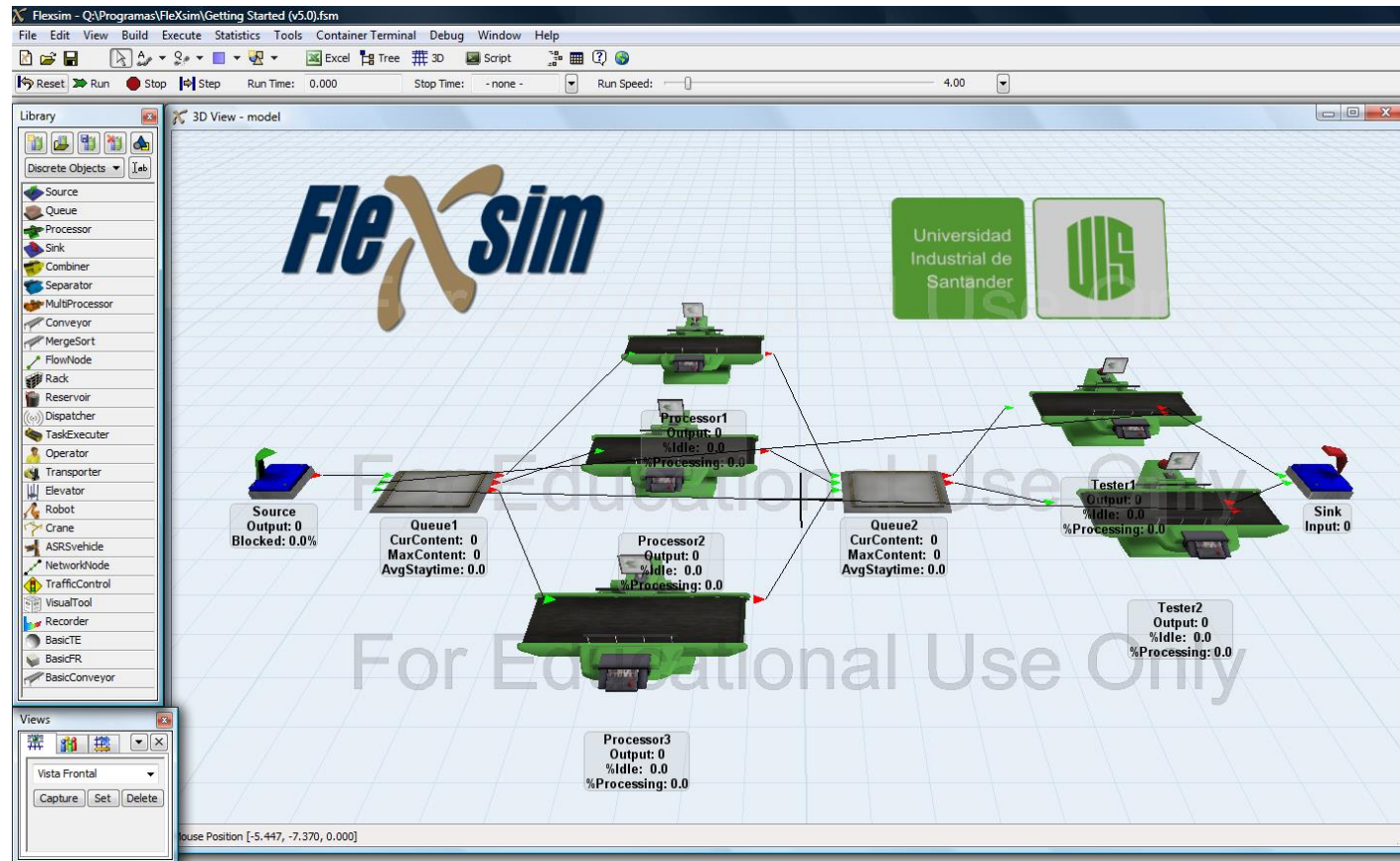
Fuente: Flexsim.

Figura 4. Warehousing o Almacenamiento



Fuente: Flexsim.

Figura 5. Ejemplo de Modelo de Simulación en Flexsim

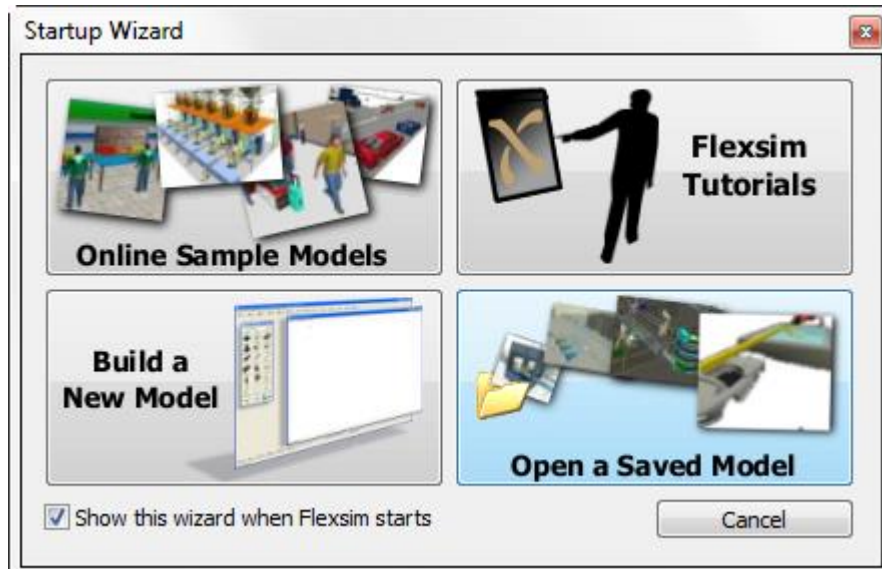


Fuente: Modelo adaptado de la Ayuda de Flexsim.

## 2. INTERFAZ GRÁFICA DE FLEXSIM

La interfaz gráfica de usuario, conocida comúnmente como *GUI* (del inglés Graphical User Interface), utiliza un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles de un programa informático en su interfaz. Su principal uso, consiste en proporcionar un entorno visual sencillo para permitir la comunicación con el sistema operativo de una máquina o computador. Al abrir el software de simulación Flexsim, el usuario se encontrará con una pantalla como la siguiente:

**Figura 6. Ventana inicial de Flexsim**



Fuente: Interfaz de Flexsim.

A través de esta ventana el usuario puede escoger alguna de las siguientes cuatro (4) opciones:

- ❖ **Build a New Model:** construir un nuevo modelo. Se requiere introducir las unidades del modelo (tiempo, longitud, fluidos).

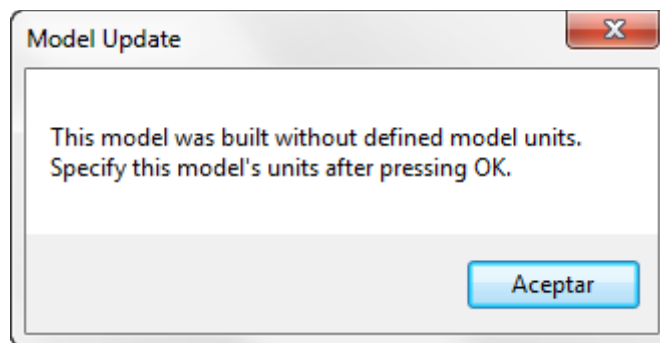
Figura 7. Unidades del modelo



Fuente: Interfaz de Flexsim.

- ❖ **Open a Saved Model:** abrir un modelo existente. Si el modelo fue creado con una versión anterior a la 6.0, aparecerá una ventana indicando que dicho modelo no tiene definidas las unidades, posteriormente visualizará la imagen anterior para seleccionar las unidades correspondientes.

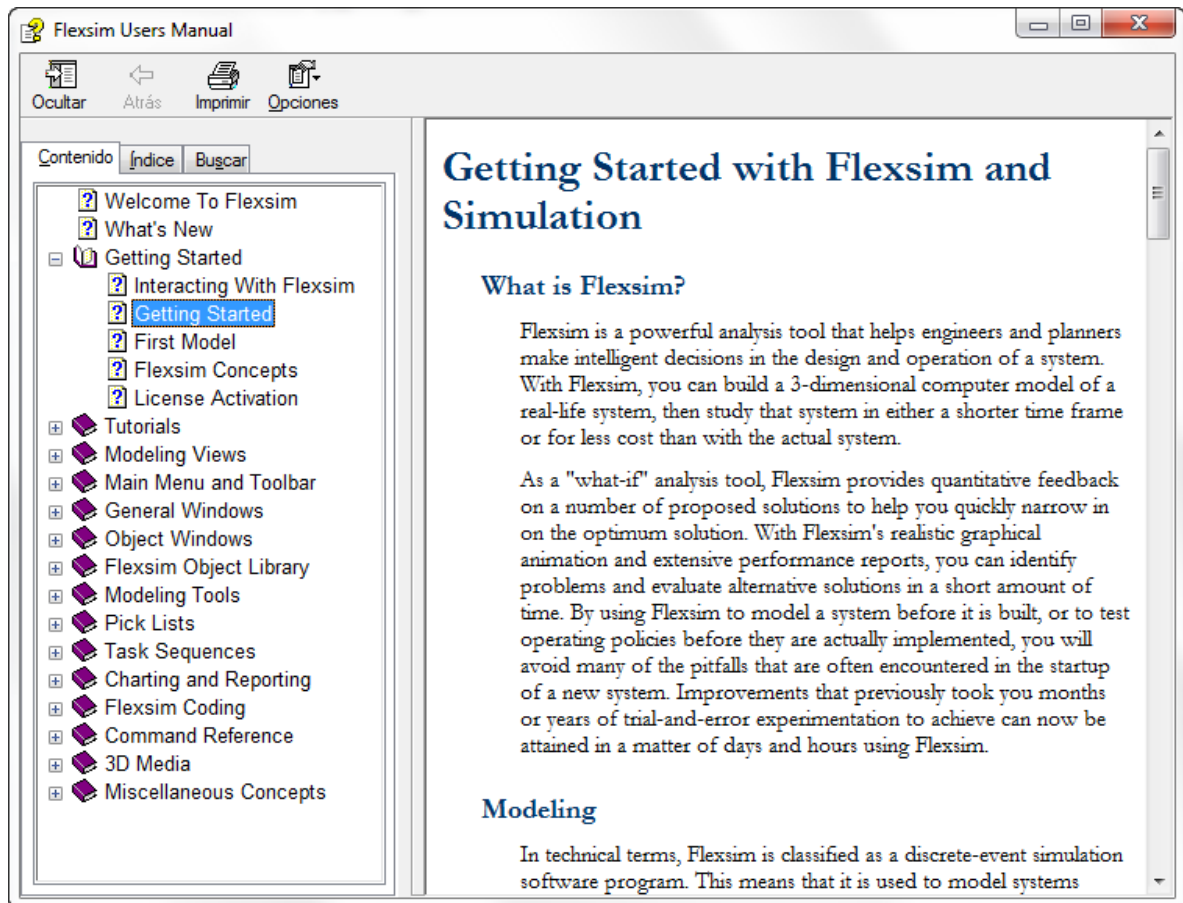
Figura 8. Actualizar modelo



Fuente: Interfaz de Flexsim.

- ❖ **Flexsim Tutorials:** abre el archivo de ayuda de Flexsim donde encontrará el tema “Tutorials” con cinco (5) tutoriales organizados en tres (3) lecciones (la segunda lección es doble) y una (1) de fluidos.

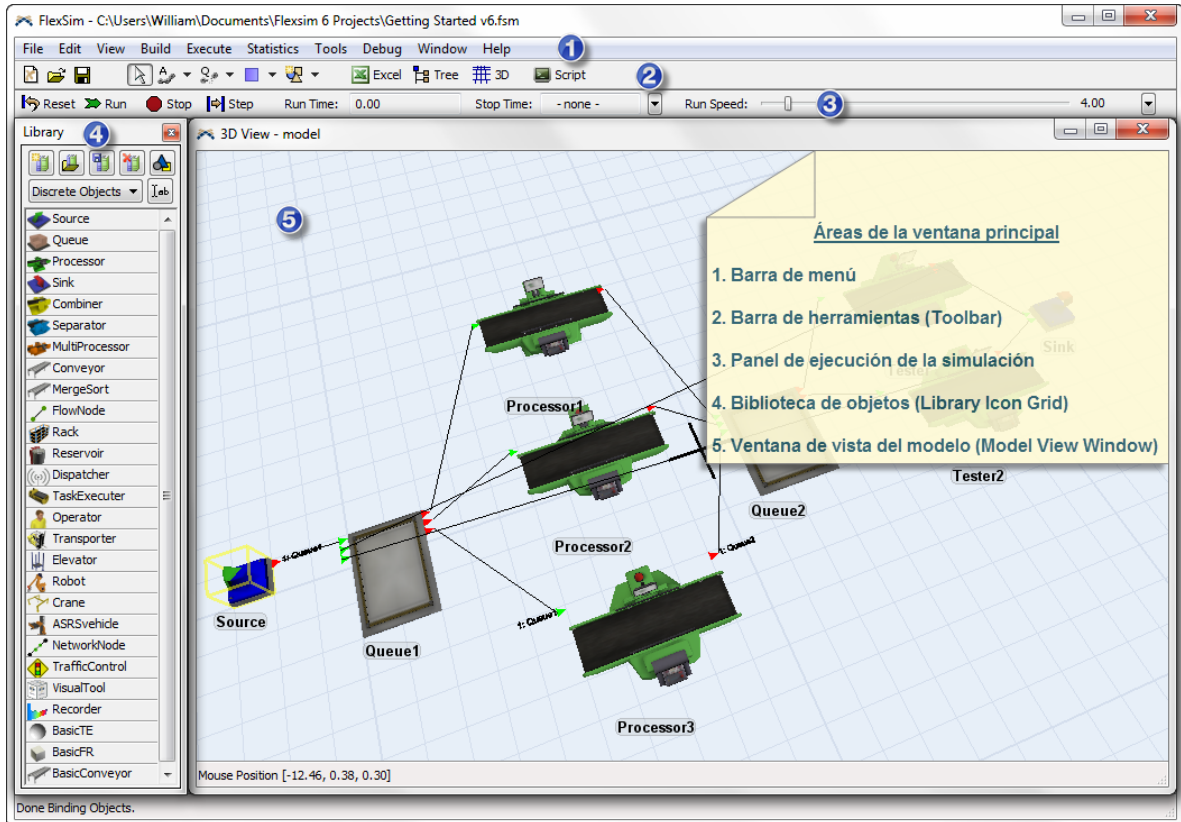
**Figura 9. Manual de usuario de Flexsim**



Fuente: Ayuda de Flexsim.

- ❖ **Online Sample Models:** simplemente abre el explorador en la página web de la comunidad de Flexsim<sup>23</sup>, donde se encuentran modelos creados por los usuarios de esta comunidad, para lo cual se requiere de una conexión a Internet.

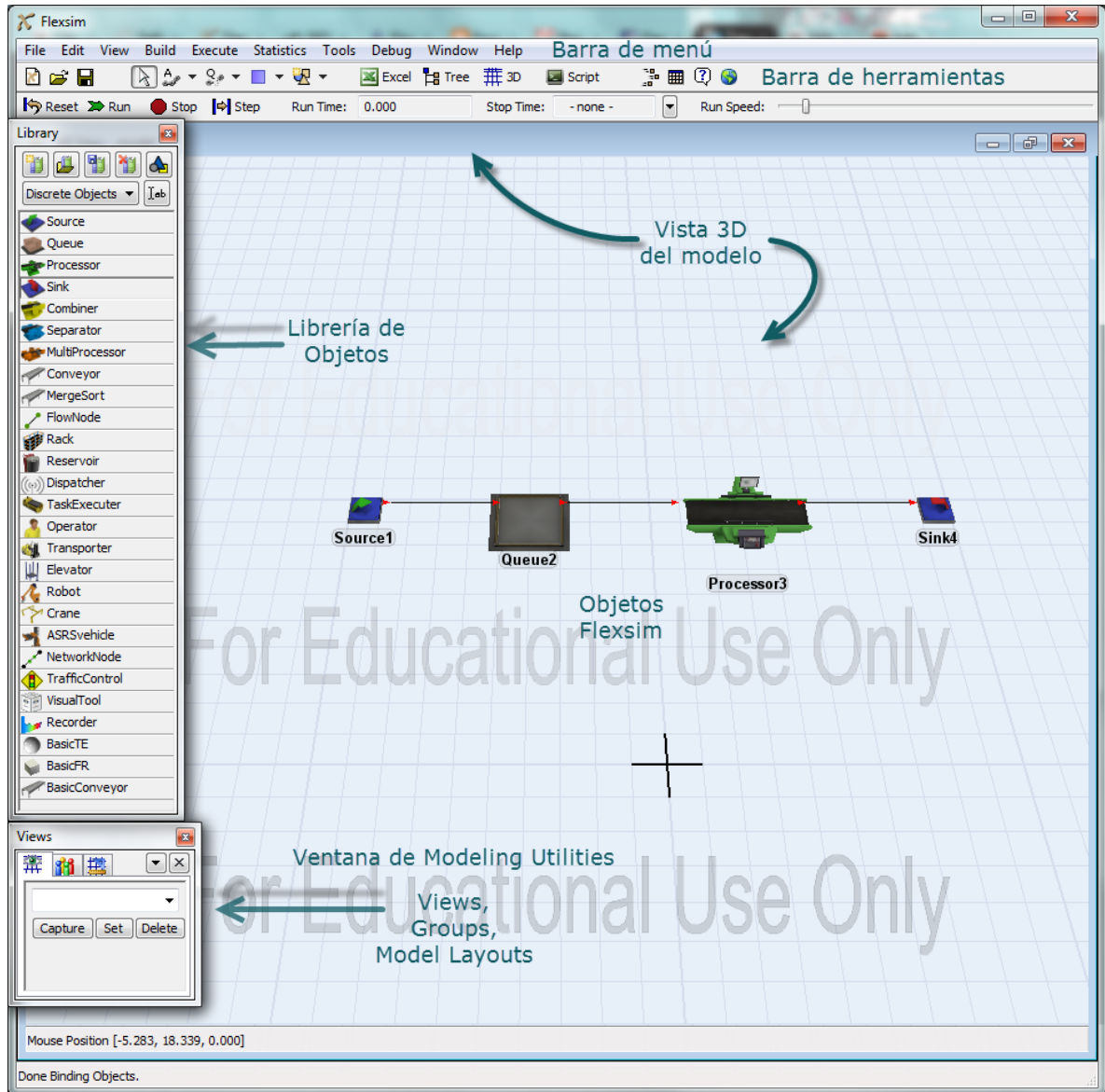
**Figura 10. Ventana principal de Flexsim**



Fuente: Autor.

<sup>23</sup> Portal Web de la comunidad de usuarios de Flexsim. Disponible en:  
<http://www.flexsim.com/community/forum/>

Figura 11. Interfaz Gráfica de Flexsim

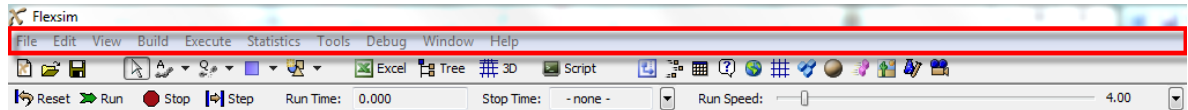


Fuente: Autor.

## 2.1 BARRA DE MENÚ

La barra de menú contiene menús desplegables como cualquier aplicación en entorno Windows.

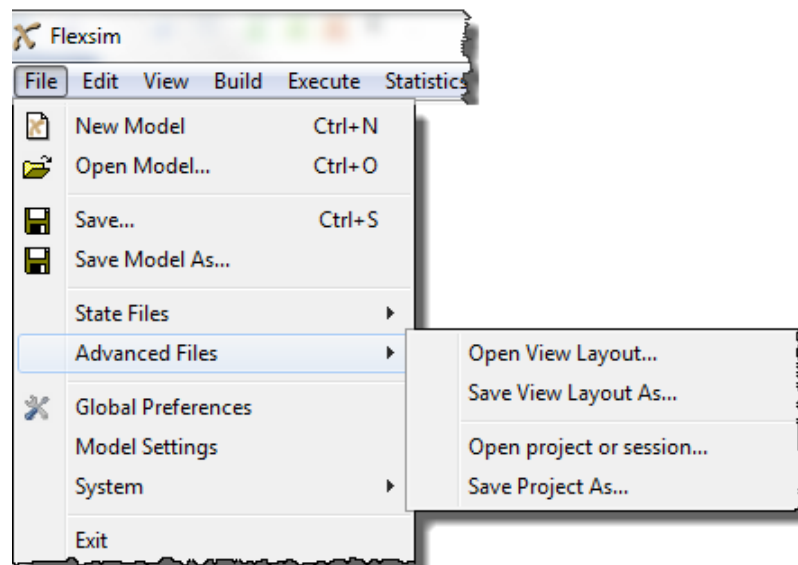
**Figura 12. Barra de menú**



Fuente: Interfaz de Flexsim.

**Menú File:** contiene las opciones para crear un Nuevo Modelo, Abrir Modelo, Guardar, Guardar como. Los modelos creados en Flexsim utilizan la extensión “.*fsm*”. Estas opciones no afectan el diseño de vista activo.

**Figura 13. Menú File | Advanced Files.**



Fuente: Interfaz de Flexsim.

Otras opciones menos utilizadas son:

- ❖ State Files o Archivos Estado: esta opción permite al usuario guardar el estado de ejecución del modelo actual, o cargar un modelo de estado guardado para continuar su ejecución. Guardar el estado es muy útil cuando el modelo se encuentra en medio de una corrida de simulación y desea guardarlo en su estado actual (todos los flowitems permanecen donde están y los recursos permanecen en su estado actual de operación), y luego cargar el estado del modelo y ser capaz de continuar ejecutando la simulación desde ese punto.
- ✓ XML Files o Archivos XML: este submenú le permite abrir y guardar archivos en formato XML en lugar del formato binario estándar Flexsim. Puede abrir archivos de modelo, proyecto, diseño de la vista, o una sesión con esta opción.

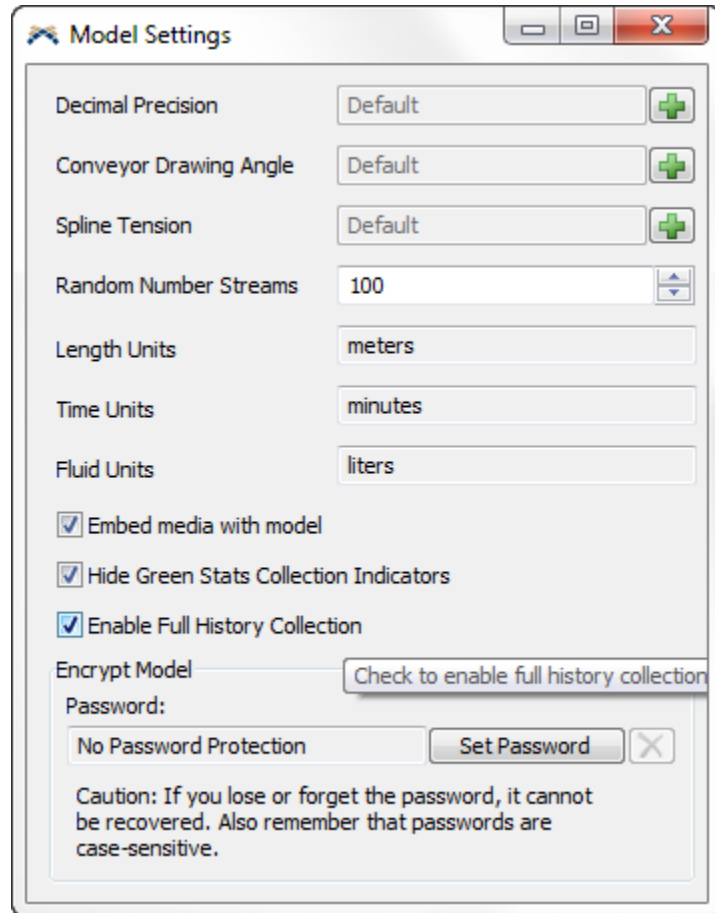
El siguiente submenú solo se encuentra disponible en la versión 5

- ❖ Advanced Files: esta menú permite al usuario cargar o guardar un diseño de vista, el proyecto, o período de sesiones (tanto diseño de vista y proyecto).
- ✓ Layout View Open - Esta opción permite al usuario elegir un archivo de vista de diseño o de presentación de Flexsim (extensión “.fsv”) para abrir. Esto va a cambiar el diseño de las ventanas en el programa, no afectará el modelo, o la librería de objetos.
- ✓ Save View Layout - Guardar un archivo de vista de presentación de Flexsim (extensión “.fsv”). Este diseño o configuración de las ventanas de Flexsim se guardará en la vista de árbol o Tree de Flexsim.
- ✓ Open a Project or Session - Permite al usuario elegir un proyecto de Flexsim o archivo de sesiones (extensión “.fsp” o “.fss”) para abrir.

Cada vez que se abre un proyecto o una sesión, se tendrá que recopilar toda la sesión en el menú View, en lugar de pulsar el botón de compilación por defecto. El botón de compilación por defecto sólo compila el código en el modelo, mientras que la opción de compilación de sesión entera en la el menú *View | System Maintenance* compilará tanto la librería como el modelo.

- ✓ Save Project As - Guardar un archivo de proyecto (".fsp"). Un archivo de proyecto se guardará en el árbol principal que incluye el sistema, la librería y el modelo.
- ✓ Save Session As - Permite elegir un archivo de sesión (extensión ".fss") para guardar o abrir. Esto reemplazará el diseño de la vista actual, el modelo, la biblioteca, y otra información global. Una sesión guarda el proyecto y la vista en un solo archivo.
- ❖ Global Preferences: esta opción abre la ventana de preferencias, lo que le permite configurar las preferencias de Flexsim. También puede configurar el texto, fuentes y colores empleados por el software, asignar o cambiar las unidades por defecto para nuevos modelos, personalización de la barra de herramientas, opciones avanzadas de compilación y gráficos.
- ❖ Model Settings: configuración del modelo actual. Se puede ajustar la precisión decimal, las semillas de números aleatorios, también se visualizan las unidades del modelo, más no se pueden cambiar desde esta ventana, se puede activar la opción *Enable Full History Collection* para recolectar todas las estadísticas del modelo para su posterior análisis con la herramienta *Flexsim Chart*. También se puede asignar una contraseña al modelo para evitar su edición.

Figura 14. Model Settings



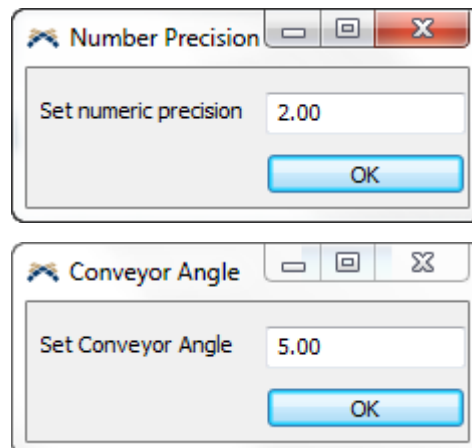
Fuente: Interfaz de Flexsim.

- ❖ System: permite recargar manualmente los medios de comunicación, así como desconectar los archivos DLL que se pueden vincular a los nodos.
- ❖ Exit: esta opción cerrará Flexsim sin guardar ningún archivo. Si el usuario desea guardar los cambios, el archivo correspondiente debe ser guardado antes de seleccionar esta opción.

**Menú Edit:** contiene las opciones de deshacer, rehacer. También se encuentran las siguientes opciones:

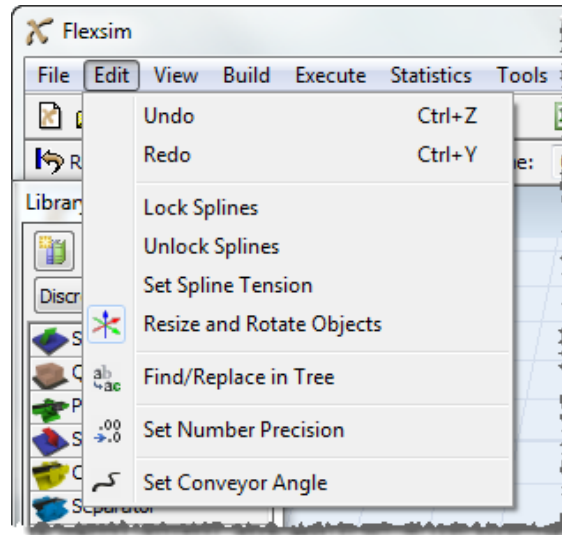
- ❖ **Resize and Rotate Objects:** esta opción es para cambiar el tamaño y rotar los objetos. Al clicar en esta opción se visualizan los ejes del objeto al seleccionarlo, en caso de no mostrarse los ejes es porque se desactivó la opción, para lo cual nuevamente se da clic en Resize and Rotate Objects.
- ❖ **Find / Replace in Tree:** esta opción abre el diálogo Buscar y Reemplazar en la estructura de árbol de Flexsim.
- ❖ **Set Number Precision:** esta opción abre un cuadro de diálogo para establecer el número de precisión. Este es el número de decimales que se muestran para los números en la sesión actual de Flexsim.
- ❖ **Set Conveyor Angle:** esta opción determina el ángulo de las secciones curvas del objeto Conveyor o cinta transportadora. La modificación de este valor solo es a nivel gráfico, no afecta los resultados del modelo.

**Figura 15. Ajuste de Number Precision y Conveyor Angle**



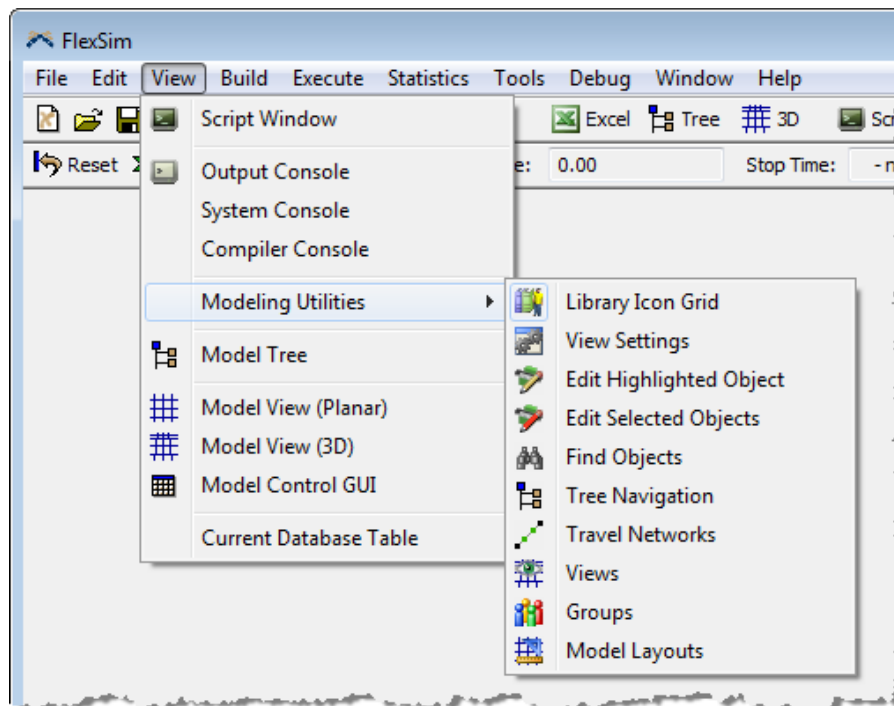
Fuente: Interfaz de Flexsim.

Figura 16. Menú Edit



**Menú View:** permite acceder a las vistas del modelo, al árbol de nodos de Flexsim y consolas que tiene la aplicación.

Figura 17. Menú View

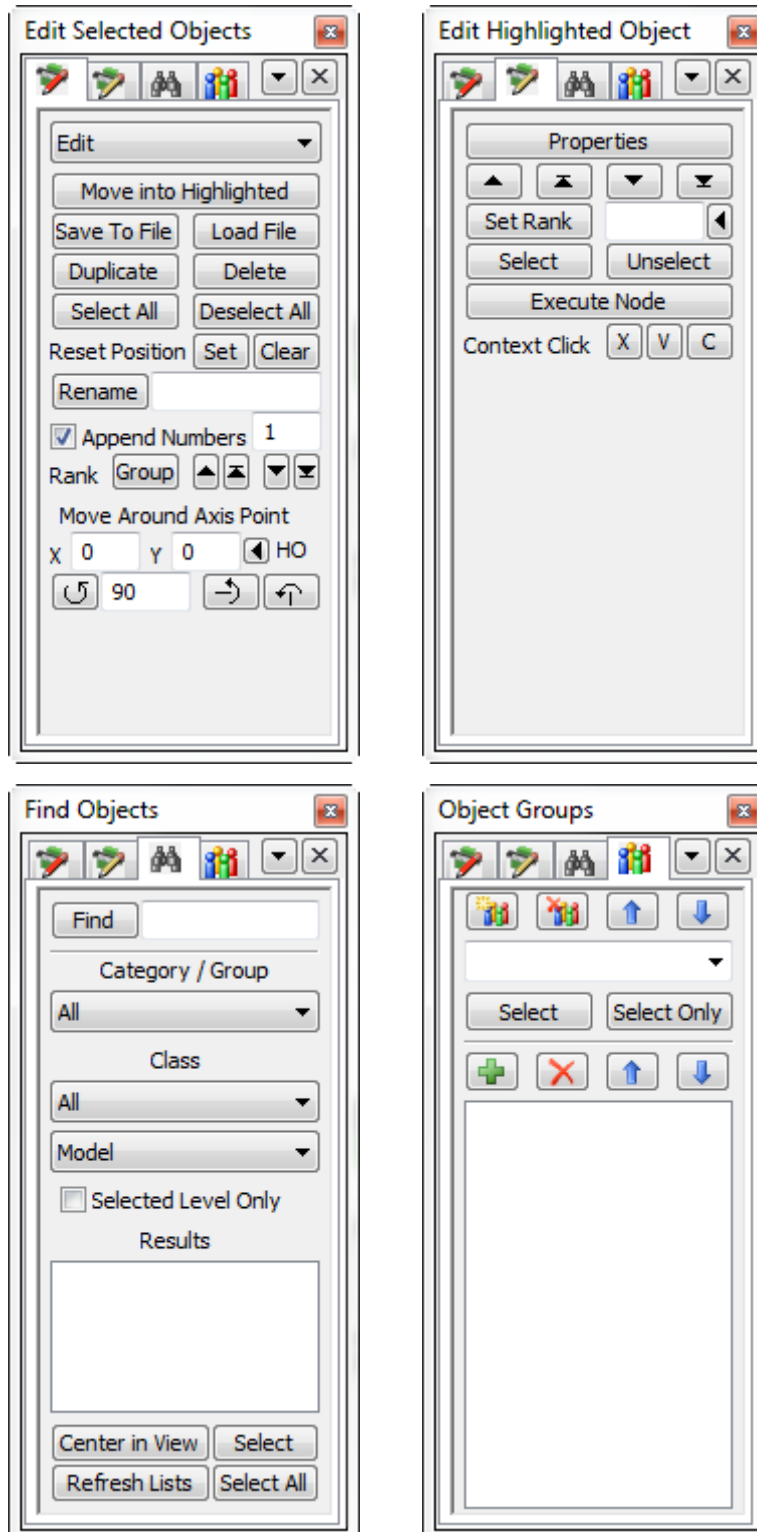


Fuente: Interfaz de Flexsim.

- ❖ Script Window o Ventana Script: esta opción abre la ventana de secuencias de comandos. La ventana de secuencias de comandos es una ventana interactiva para ejecutar comandos Flexscript manualmente. Es útil para probar comandos que no son tan comunes y ver su resultado.
  
- ❖ Output Console o Consola de Salida: esta opción abre una ventana donde se muestra la información de salida. Usted puede imprimir su propia información a la salida de la consola con los comandos: PT (), PR (), PD (), PF (), etc. Para más información sobre estos comandos, consulte el archivo de ayuda de Flexsim.
  
- ❖ System Console o Consola del sistema: esta opción abre una ventana donde se imprime información sobre el estado del motor del programa. Si esta ventana está abierta, el modelo puede moverse muy lentamente.
  
- ❖ Compiler Console o Compilador: esta opción abre una ventana donde se imprime la información, mientras que el modelo se compila. Esto muestra el estado de cada paso del proceso de compilación. Esta consola también muestra los errores Flexscript cuando se genera código Flexscript.
  
- ❖ Model Tree o Estructura de Árbol del Modelo: esta opción abre una ventana de árbol que muestra la carpeta del modelo, esta ventana visualiza en nodos todos los objetos que están en el modelo y su respectiva programación y codificación interna. El árbol puede ser manipulado desde esta vista.
  
- ❖ Model View o Vista de Modelo (Planar): esta opción abre una vista planar del modelo que muestra el modelo en dos dimensiones. Las formas 3D de los objetos no se dibujan en esta vista.

- ❖ Modelo View (3D) o Vista de Modelo 3D: esta opción abre una vista 3D del modelo.
  
- ❖ Current Database Table o Tabla de Base de Datos Actual - Esta opción abre una ventana que muestra la tabla de base de datos activa que fue abierta o consultada con el comando *dbopen()*, *dbchangetable()*, *dbsqlquery()*.
  
- ❖ Modeling Utilities o Utilidades de modelado
  - ✓ Icon Library Grid – Abre la librería de objetos de Flexsim en una ventana que contiene los iconos de los objetos que se pueden arrastrar en el modelo.
  
  - ✓ View Settings - Esta opción abre la configuración de vista.
  
  - ✓ Edit Highlighted Object - Edita el objeto resaltado que tiene el contorno amarillo.
  
  - ✓ Edit Selected Objects – Edita los objetos seleccionados, es decir, aquellos que tienen contorno rojo.
  
  - ✓ Find Objects - Esta opción abre la ventana Buscar objetos, por categoría y clases.
  
  - ✓ Groups - Esta opción abre una utilidad que permite agrupar varios objetos asignándole un nombre a ese grupo creado, para su clasificación, búsqueda y edición. Se puede agrupar por áreas, zonas, clase de objeto, servicio, máquina, proceso, etc.

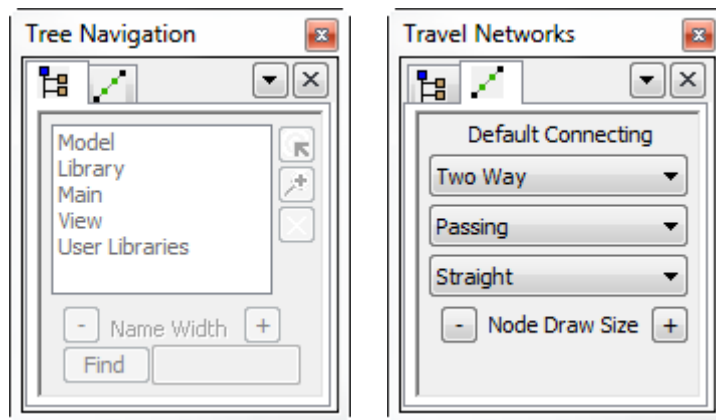
Figura 18. Utilidades del submenú Modeling Utilities



Fuente: Interfaz de Flexsim.

- ✓ Tree Navigation - Árbol de navegación - Esta opción abre la utilidad de navegación de árbol de nodos de Flexsim donde se encuentra condensado de forma nodular toda la estructura de Flexsim y el modelo. Accesible únicamente en versiones licenciadas, requiere un uso más avanzado.
- ✓ Travel Networks - Esta opción abre la utilidad de las redes de transporte.

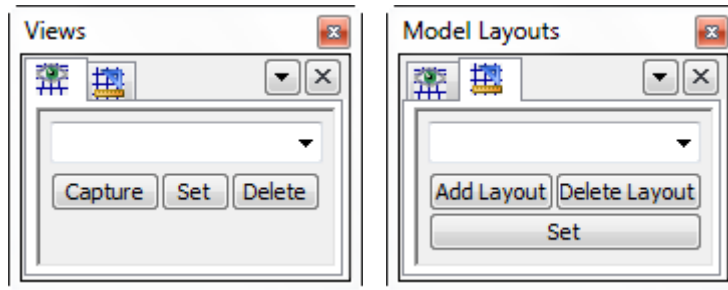
**Figura 19.** Utilidades Tree Navigation y Travel Networks



Fuente: Interfaz de Flexsim.

- ✓ Views - Vistas - Esta opción abre la utilidad de vistas, donde se pueden crear vistas personalizadas de áreas de interés de la simulación.
- ✓ Model Layouts - Esta opción abre una ventana donde se pueden definir diferentes configuraciones de distribución de los objetos de Flexsim, útil para probar escenarios con diferentes distribuciones de planta.

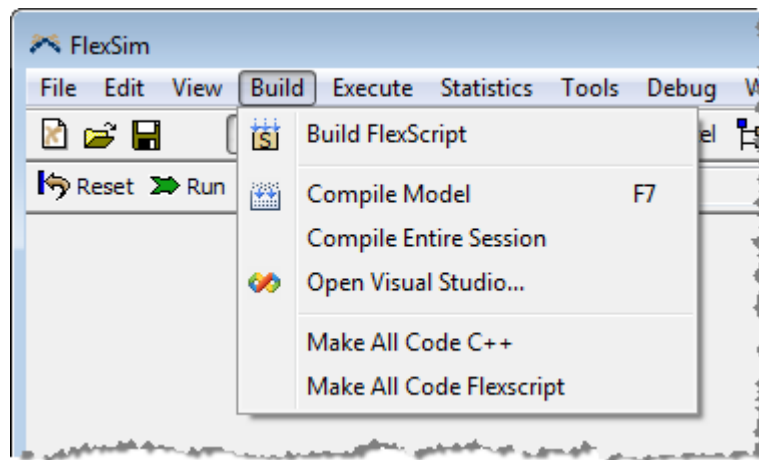
**Figura 20. Utilidades View y Model Layouts**



Fuente: Interfaz de Flexsim.

**Build:** menú con opciones para compilar los modelos realizados y crear código de programación en lenguaje C++ o Flexscript, que es el lenguaje propio del software Flexsim.

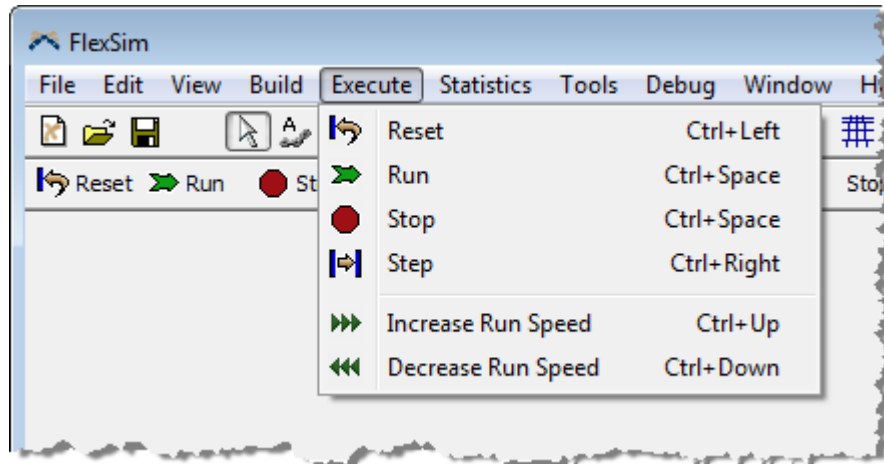
**Figura 21. Menú Build**



Fuente: Interfaz de Flexsim.

**Execute:** menú con comandos de ejecución de la simulación como *Reset* para reiniciar el modelo, *Run* para correr la simulación, *Stop* para detenerla, *Step* para ejecutarla paso a paso y los dos últimos comandos para aumentar o disminuir la velocidad de corrida de la simulación.

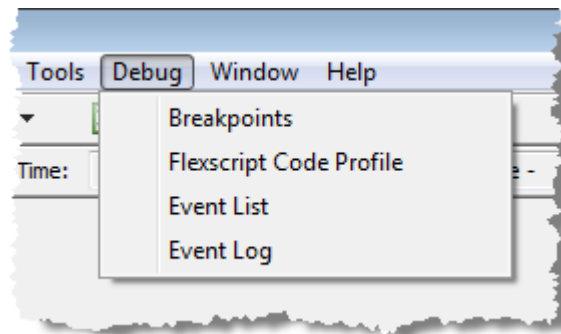
Figura 22. Menú Execute



Fuente: Interfaz de Flexsim.

**Debug:** menú de depuración de la simulación. Con la opción *Breakpoints* creamos puntos de ruptura para ejecutar o correr la simulación hasta que alcance dicho *Breakpoint*.

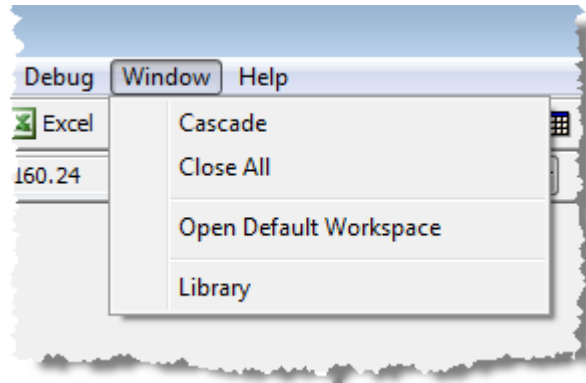
Figura 23. Menú Debug



Fuente: Interfaz de Flexsim.

**Window:** menú con opciones para organizar las ventanas activas de la pantalla principal.

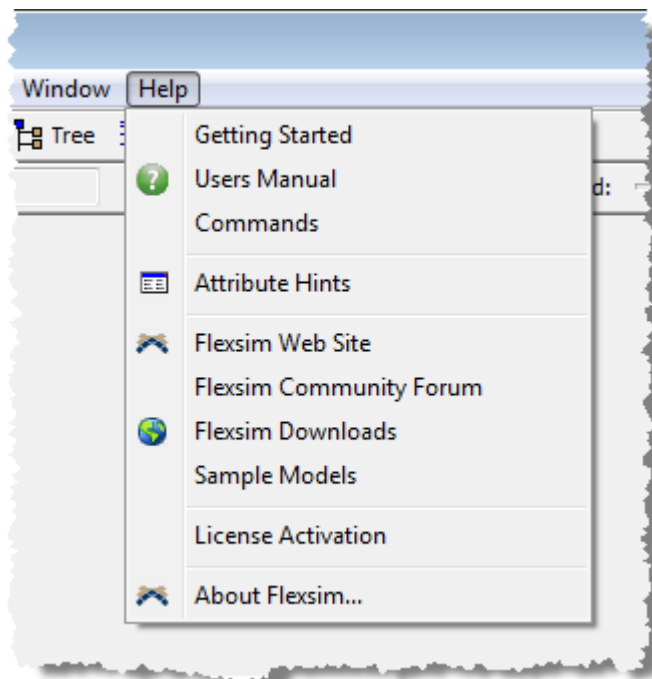
Figura 24. Menú Window



Fuente: Interfaz de Flexsim.

**Help:** visualiza las opciones de ayuda de Flexsim, tales como la guía de usuario, la lista de comandos, enlaces al portal web de Flexsim y la comunidad de usuarios y la ventana de activación de la licencia.

Figura 25. Menú Help

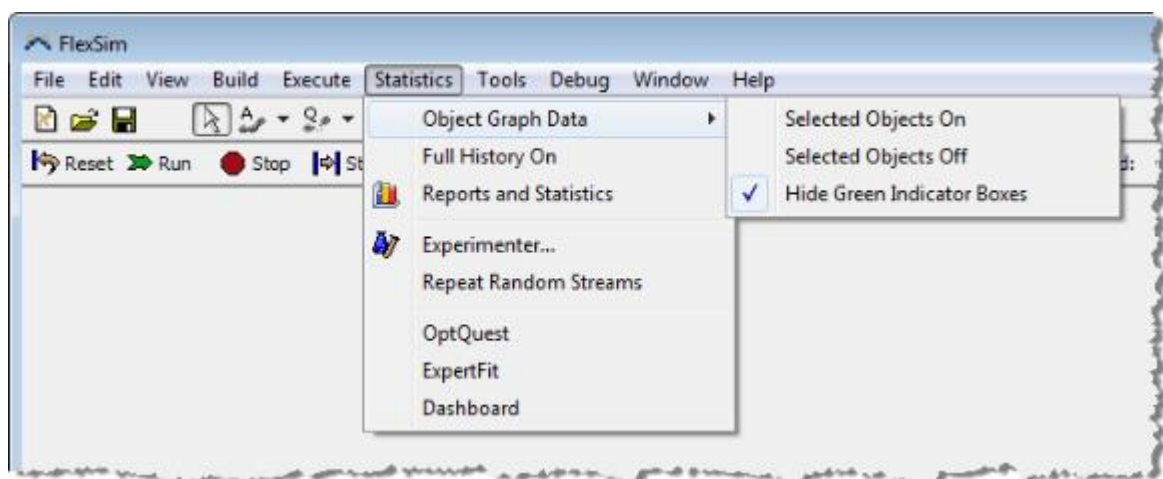


Fuente: Interfaz de Flexsim.

**Statistics:** en este menú se encuentran utilidades para análisis y recogida de datos, para lo cual es necesario activar la opción *Full History On*. Se encuentran las siguientes herramientas:

- ❖ **Experimenter:** esta opción permite realizar diseños experimentales del modelo, a través de múltiples escenarios de ejecución.
- ❖ **OptQuest:** esta opción abre la herramienta de optimización para encontrar los valores óptimos para un conjunto de variables de decisión a fin de minimizar o maximizar una función objetivo dentro de los límites definidos por el usuario.
- ❖ **ExpertFit:** es un programa usado para determinar la mejor distribución de probabilidad estadística que se ajusta a los datos de entrada, mediante pruebas de bondad de ajuste como: Kolmogorov-Smirnov, Anderson-Darling y Chi cuadrado.
- ❖ **Dashboard:** esta opción abre un panel de control, donde puede agregar gráficos y otra información acerca de la salida del modelo.

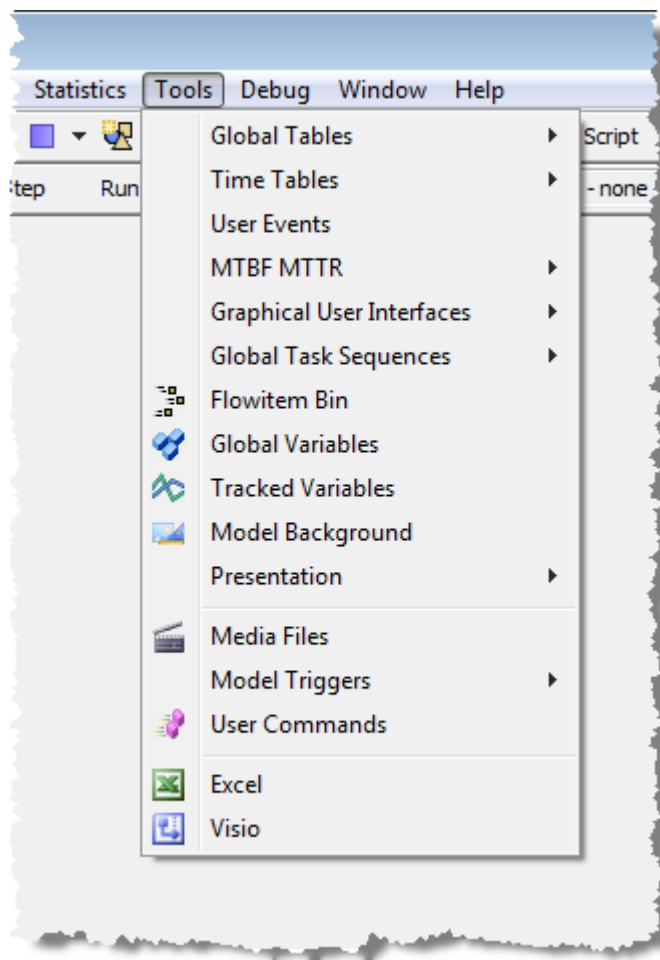
**Figura 26. Menú Statistics**



Fuente: Interfaz de Flexsim.

**Tools:** menú con herramientas para el desarrollo y programación del modelo, tales como la definición de variables, atributos, creación de flowitems, Interfaces Gráficas de Usuario (GUIs), secuencias de tareas, eventos de usuario, creación y edición de tablas globales, tablas de tiempo, tiempos de falla y reparación, controladores o disparadores globales del modelo, importación desde herramientas ofimáticas como Excel y Visio.

**Figura 27. Menú Tools**

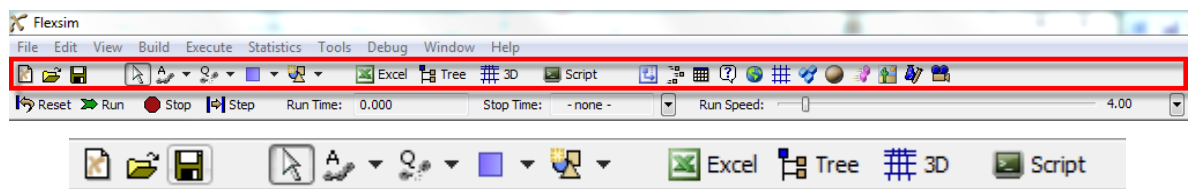


Fuente: Interfaz de Flexsim.

## 2.2 BARRA DE HERRAMIENTAS

Llamada en inglés *toolbar*, es un componente de una interfaz gráfica de usuario mostrada usualmente en pantalla a modo de fila, columna, o bloque que contiene iconos que, al ser presionados, activan ciertas funciones de una aplicación. Se presenta debajo de la barra de menú y contiene los botones más utilizados por defecto de Flexsim.

**Figura 28. Barra de herramientas (toolbar)**



Fuente: Interfaz de Flexsim.

Si se desconoce la acción asociada a los botones de la barra de herramientas, basta con ubicarse encima de ellos sin pulsar ningún clic para que aparezca un texto descriptivo, como se observa en la imagen a continuación.

**Figura 29. Texto descriptivo de la toolbar**

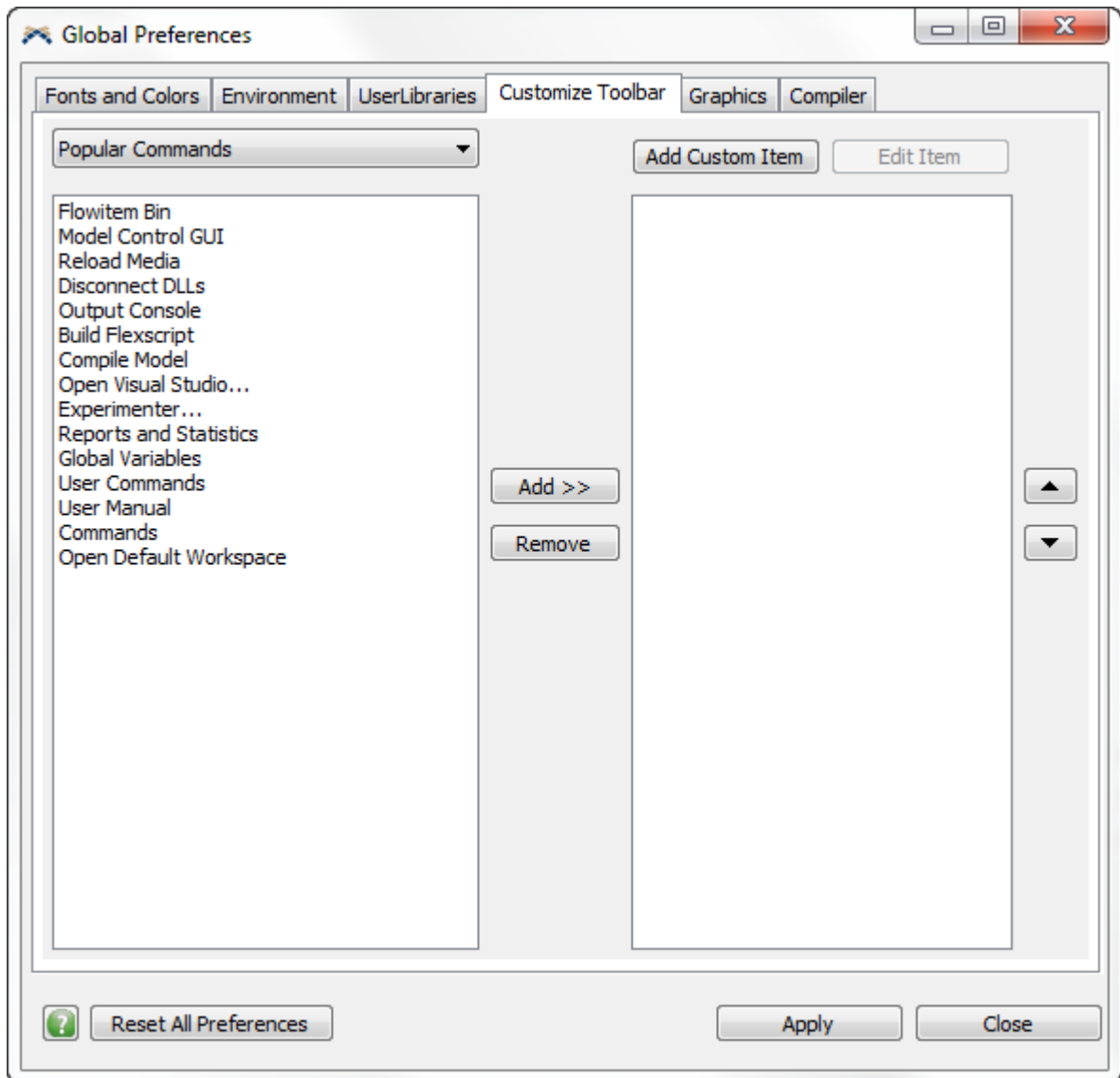


Fuente: Interfaz de Flexsim.

La barra de herramientas o toolbar de Flexsim es fácilmente configurable al igual que la mayoría de aplicaciones informáticas de Windows, como Office. Esta barra puede personalizarse colocando los comandos más utilizados en ella, para hacerlo, el usuario debe dirigirse al menú **File -> Global Preferences -> Pestaña Customize Toolbar**, posteriormente se selecciona el comando en el panel izquierdo y clic en *Add>>* para enviarlo a la ventana de la derecha, se aplican

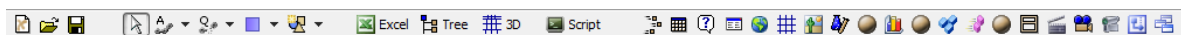
cambios (Apply) y se cierra la ventana (Close). Un comando práctico para agregar es **Global Variables** y **User Events**.

**Figura 30. Personalización de la barra de herramientas (toolbar)**



Fuente: Interfaz de Flexsim.

**Figura 31. Barra de herramientas personalizada**



Fuente: Interfaz de Flexsim.

## 2.3 PANEL DE EJECUCIÓN DE LA SIMULACIÓN

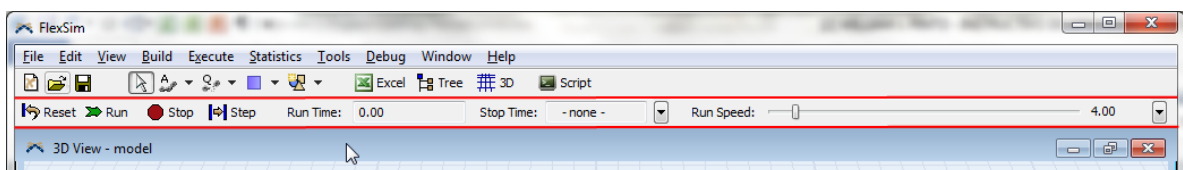
Presente debajo de la barra de herramientas. Contiene los siguientes botones para controlar la ejecución de la simulación:

- ❖ Reset: reinicia el modelo.
- ❖ Run: ejecuta la simulación.
- ❖ Stop: detiene la ejecución del modelo y el reloj de la simulación. Puede ejecutarse o correr nuevamente desde este tiempo de detención.
- ❖ Step: ejecución paso a paso de la simulación.

### Controles de tiempo:

- ❖ Run Time: visualiza el tiempo de simulación actual en las unidades de tiempo definidas en el modelo.
- ❖ Stop Time: establece un tiempo en el que la simulación será detenida (una vez parada puede establecerse un nuevo tiempo).
- ❖ Run Speed: velocidad de simulación. Define el número de unidades de tiempo de simulación por segundo de tiempo real

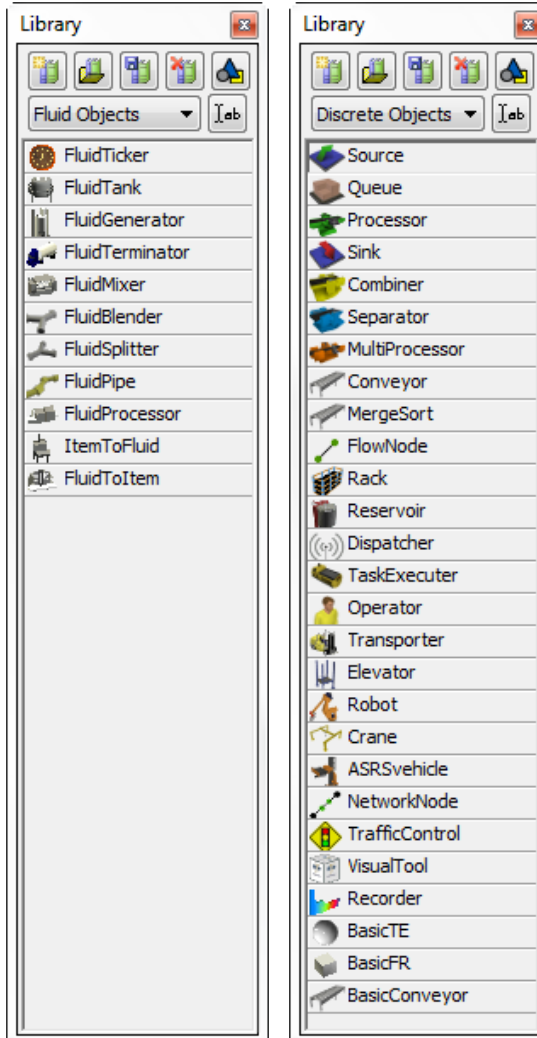
**Figura 32. Panel de ejecución de la simulación**



Fuente: Interfaz de Flexsim.

## 2.4 LIBRERÍA DE OBJETOS

Figura 33. Librería de Objetos Discretos y de Fluidos

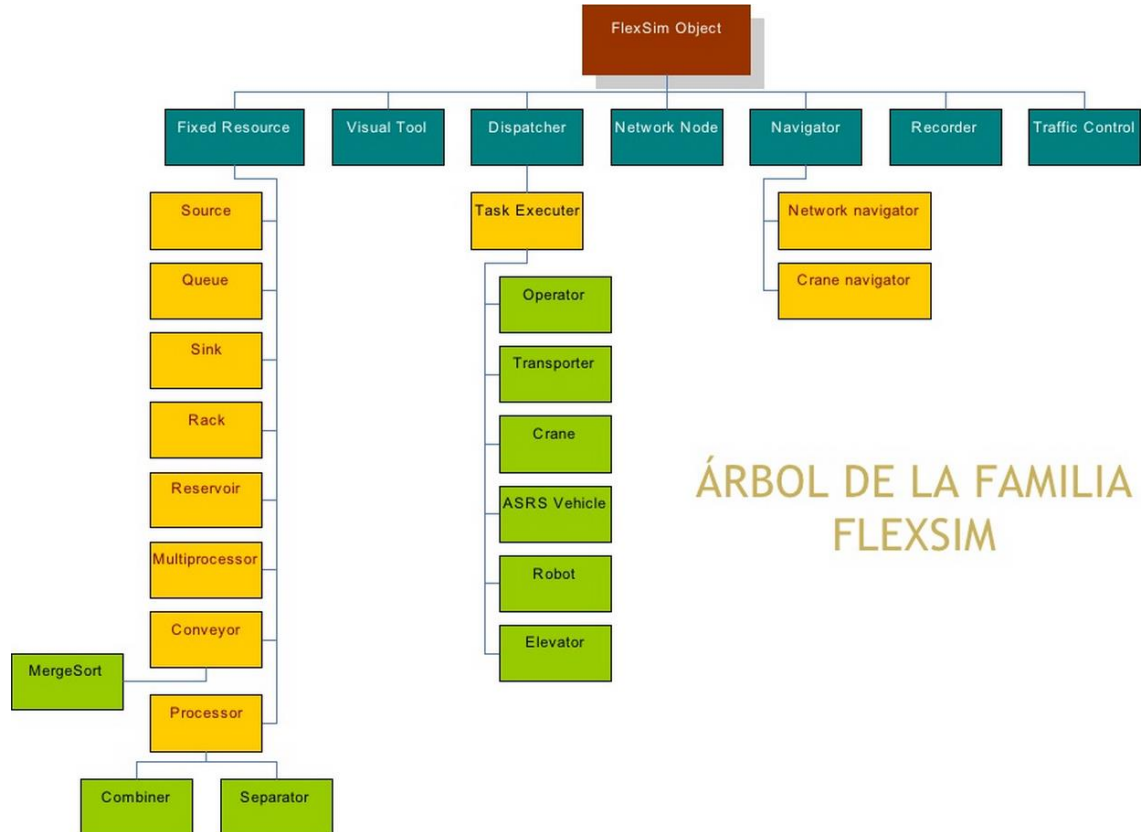


Fuente: Interfaz de Flexsim.

En la anterior figura encontramos la librería de objetos discretos y de fluidos (continuos) que se utilizan para simular el sistema de interés. Si por alguna razón no aparece esta ventana, se puede visualizar desde el menú **View**, opción **Modeling Utilities**, clic en la opción **Library Icon Grid**.

### 2.4.1 Árbol de la Familia Flexsim

Figura 34. Árbol de la familia Flexsim para objetos discretos



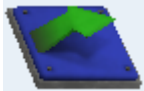
Fuente: MASINI, Javier. (2011). *Construcción de modelos usando Flexsim*. Recuperado el 10 de septiembre de 2012 en: <http://www.slideshare.net/jmasini/construccin-de-modelos-usando-flexsim>, diapositiva 11.

Este árbol de la familia Flexsim representa la clasificación de los objetos discretos que maneja Flexsim en grupos de objetos o familias dentro de una misma clase o con características similares. Por ejemplo el primer grupo de la figura anterior es el de recursos fijos o *Fixed Resources* que contiene todos los objetos para simular estaciones o localizaciones. La mayoría de objetos discretos en la librería están clasificados en una de las dos superclases de objetos en Flexsim: **Fixed Resources** y **TaskExecuters**, estos a su vez contienen como subclasses otros objetos discretos.

## 2.4.2 Objetos Discretos

Los objetos simulan diferentes tipos de recursos en la simulación. Entre los objetos principales utilizados por Flexsim tenemos los siguientes *Fixed Resources*:

### Source



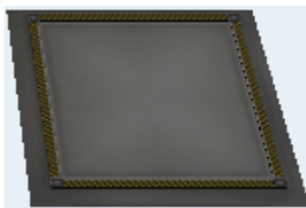
Es la fuente de entrada que crea y libera los “flowitems” (Entidades) que viajan a través del modelo. Cada “Source” crea una clase de “flowitem” y pueden asignárseles propiedades como tipo de entidad o color. Los modelos deben tener al menos un “Source” en ellos. Las fuentes pueden crear “flowitems” por una tasa entre llegadas, por una lista programada de llegadas o simplemente desde una secuencia definida de llegadas.

### Sink



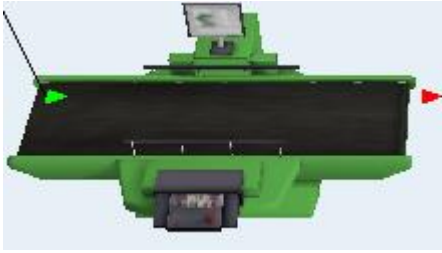
Es usado para destruir los “flowitems” o darles salida del sistema. Una vez que un “flowitem” llega al “Sink” (Salida) este no puede ser recuperado para su reingreso en el sistema. Cualquier recolección de datos que envuelve algún “flowitem” que va a dejar el sistema debe hacerse justo antes de entrar al “Sink” o en la pestaña del trigger: “OnEntry” del “Sink”.

### Queue



La cola es un buffer o área de almacenamiento usada para mantener los “flowitems” (entidades) cuando un objeto no puede ser aceptado por un proceso posterior. Por defecto la cola trabaja con disciplina FIFO (First In First Out) o PEPS. El “Queue” tiene opciones para acumular múltiples “flowitems” antes de que los suelte para entregarlos a los objetos posteriores o hasta que complete su capacidad máxima.

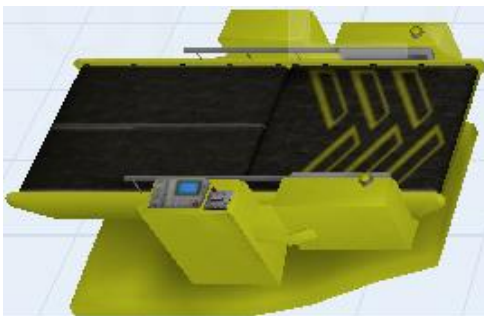
## Processor



Es usado para simular el procesamiento de “flowitems” en un modelo. El proceso es simplemente modelado como un retraso forzado de tiempo. El tiempo total de proceso está dividido entre el tiempo de setup y el tiempo de proceso.

El “Processor” puede procesar más de un “flowitem”, a la vez. Además puede ser determinada la falla y reparación de un procesador a intervalos regulares o aleatorios. Los “Processor” pueden llamar por operadores durante sus tiempos de setup, proceso y/o reparación. Cuando un procesador falla, todos los “flowitems” que está procesando se retrasarán. Puede procesar múltiples flowitems independientemente (si su capacidad máxima es mayor que 1).

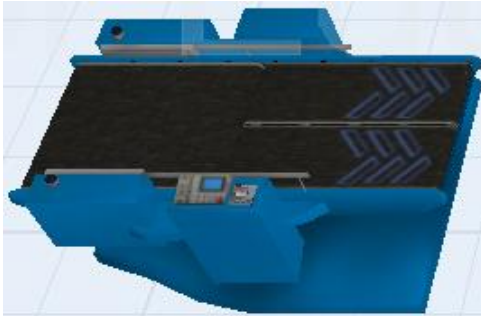
## Combiner



El “Combiner” se utiliza para agrupar varios “flowitems” en uno solo para viajar a través del modelo. Esta unión puede ser permanente (Join), o pueden ser empacados para que puedan ser separados en un momento posterior en el tiempo (Pack). El “Combiner” en primer lugar recibe un solo flowitem a través de su primer puerto de entrada, que va actuar como entidad base o contenedor de los flowitems que van a ser agrupados con este de forma temporal o permanente. Posteriormente recibe los flowitems a ser combinados a través de los puertos de entrada restantes, es decir, del input port 2 y superiores, especificando la cantidad de flowitems a recibir.

Después que todos los flowitems requeridos por el usuario arriben al *combiner* el tiempo de setup / proceso comienza. El “Combiner” se puede configurar para utilizar operadores durante sus tiempos de setup, proceso y/o reparación.

## Separator



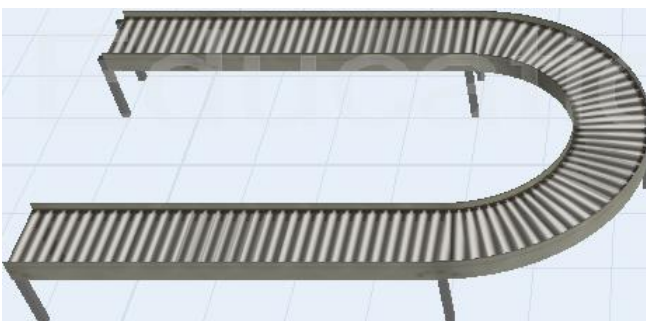
El “Separator” se utiliza para separar un flowitem en múltiples partes. Esta separación puede ser por un desempaque (Unpack) de un flowitem que ha sido empacado previamente en un Combiner o por la realización de múltiples copias del flowitem original (Split) para simular un proceso de corte. La separación de entidades es hecha después que el tiempo de proceso ha terminado. El *Separator* se puede configurar para utilizar operadores durante sus tiempos de setup, proceso y/o reparación.

## MultiProcessor



Es usado para simular el procesamiento de *flowitems* en operaciones ordenadas secuencialmente. El usuario define un conjunto de procesos para cada *MultiProcessor* con diferentes tiempos y llamadas por operadores separados durante las etapas de sus operaciones. Puede procesar un flowitem a la vez.

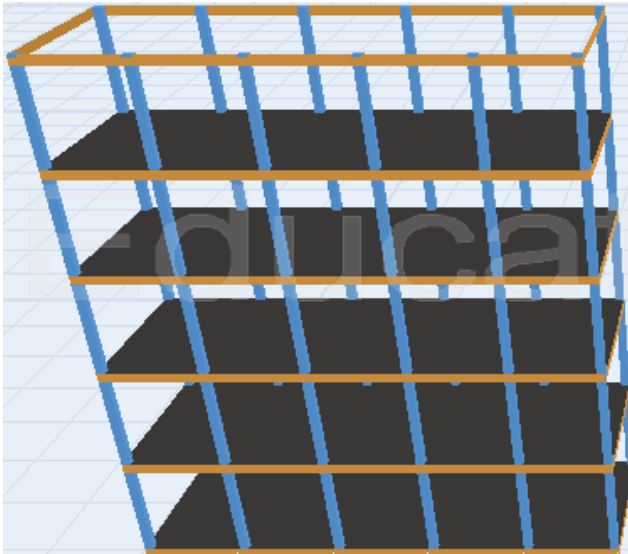
## Conveyor



El conveyor es una banda o cinta transportadora que se utiliza para mover flowitems a través del modelo a lo largo de un camino. El trayecto se define mediante la creación de diferentes secciones de la cinta transportadora. Cada sección puede ser recta o curva. Las secciones

curvas se definen por el ángulo de curvatura en grados y su radio. Las secciones rectas se definen por su longitud. Esto permite que el conveyor tenga tantas curvas y codos como se requiere. El conveyor puede ser de acumulación o no acumulación.

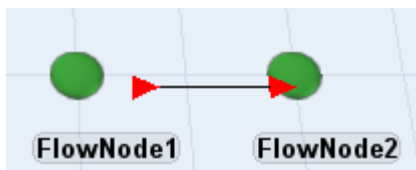
### Rack



Un “Rack” es utilizado para almacenar flowitems. El número y tamaño de las bahías y niveles puede ser definido por el usuario. También se puede especificar la bahía y el nivel para colocar flowitems que entran. Si un objeto transportador se utiliza para entregar o recoger flowitems de un rack, el transporte se dirigirá a la celda específica en el rack, donde el

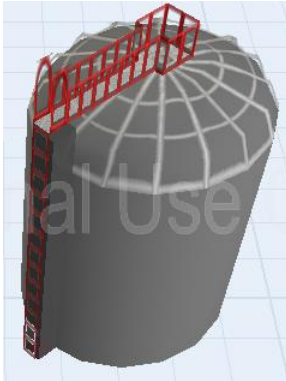
flowitem está asignado. También puede ser utilizado como almacenamiento en el suelo de un almacén, utilizando el número de bahía para especificar una posición x para colocar un flowitem en el suelo, y el nivel para especificar la posición y para colocar el flowitem.

### FlowNode



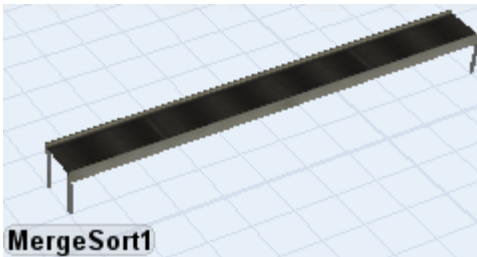
Son recursos fijos que pueden actuar de una manera similar a una muy simple cinta transportadora de acumulación. Los flowitems se mueven a lo largo de las conexiones de salida de un nodo de flujo pero sin ninguna pieza visible de equipo físico.

## Reservoir



Es un depósito que se utiliza para almacenar flowitems como si estuvieran en un silo de almacenamiento o un tanque de fluido. La tasa de flujo entrante y saliente del depósito puede ser definida por el usuario. Los eventos pueden ocurrir cuando el nivel del depósito sube o baja a ciertos valores que el usuario puede definir.

## MergeSort



El MergeSort es una banda o cinta transportadora de no acumulación que permite tener múltiples posiciones de entrada, así como también múltiples posiciones de salida a lo largo de la banda. Cada puerto de entrada / salida de la cinta transportadora tiene una entrada / salida definida por el usuario.

- ❖ **TaskExecuters:** son los recursos móviles que transportan flowitems de un lugar a otro. TaskExecuter hace referencia a una superclase de objetos y también a un objeto llamado de la misma manera, por lo tanto para evitar esta confusión, de aquí en adelante cuando se mencione solamente TaskExecuter hará referencia a la superclase que contiene varios objetos en subclases, de otra manera se mencionará: "...el objeto TaskExecuter".

A continuación se describen algunos recursos móviles o ejecutores de tareas:

## Operator



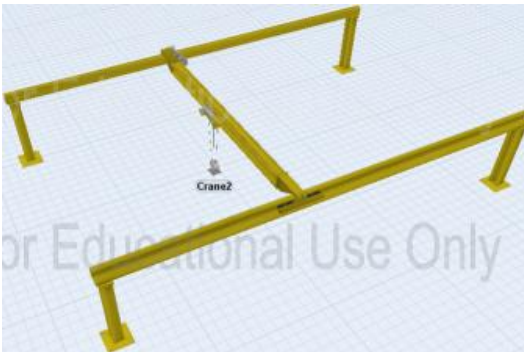
Aparte de realizar transportes de *flowitems* desde un objeto a otro, puede realizar operaciones en las estaciones o FixedResources. Puede personalizarse para llevar múltiples “flowitems”. Un único transporte puede recibir múltiples solicitudes por tareas o trabajos (tal como transportar un *flowitem*) de varios objetos localizados en el modelo.

## Transporter



Es utilizado principalmente para llevar “flowitems” desde un objeto a otro. Puede personalizarse para llevar múltiples “flowitems”. Un único transporte puede recibir múltiples solicitudes por tareas o trabajos (tal como transportar un “flowitem”) de varios objetos localizados en el modelo. Esas tareas son colocadas en una cola para el “transporter”. Cuando un “transporter” termina una tarea, este inmediatamente comenzará la siguiente tarea en su cola o esperará hasta que la próxima tarea sea enviada a ésta.

## Crane



La grúa tiene una funcionalidad similar a la del transportador, pero con un gráfico modificado. Trabaja en un espacio fijo siguiendo movimientos rectangulares x, y, z. Por defecto, el selector de la grúa se eleva a la altura del objeto de la grúa después de recoger o dejar a un *flowitem* antes de que se traslade a la siguiente ubicación.

Algunos modelos requerirán un “dispatcher” para poner en cola las solicitudes por trabajo, y otros modelos requerirán el transporte o el operador de forma individual para poner en cola las tareas.

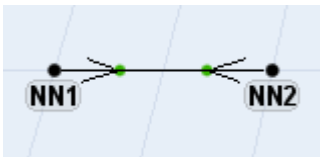
- ❖ **Otros objetos discretos:** a continuación se describen los objetos que no están dentro de las dos superclases anteriores:

### Dispatcher



Es usado para controlar un grupo de transportes u operadores. Las peticiones por trabajo son enviadas al “dispatcher” desde un objeto y el “dispatcher” los delega a los transportes u operadores que está controlando. La tarea será realizada por el transporte u operador que finalmente recibe la solicitud.

### NetworkNode



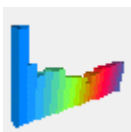
Son nodos utilizados para crear redes de transporte para las entidades. Indican por donde debe circular una entidad o un transporte. Se conectan tanto a estaciones fijas como a objetos o recursos móviles.

### VisualTool



Objeto utilizado para decorar el modelo con un gráfico en 3D, fondos, visualizar texto, para darle un aspecto más realista y estético. Otro uso del “VisualTool” es como un contenedor de otros objetos en el modelo.

### Recorder

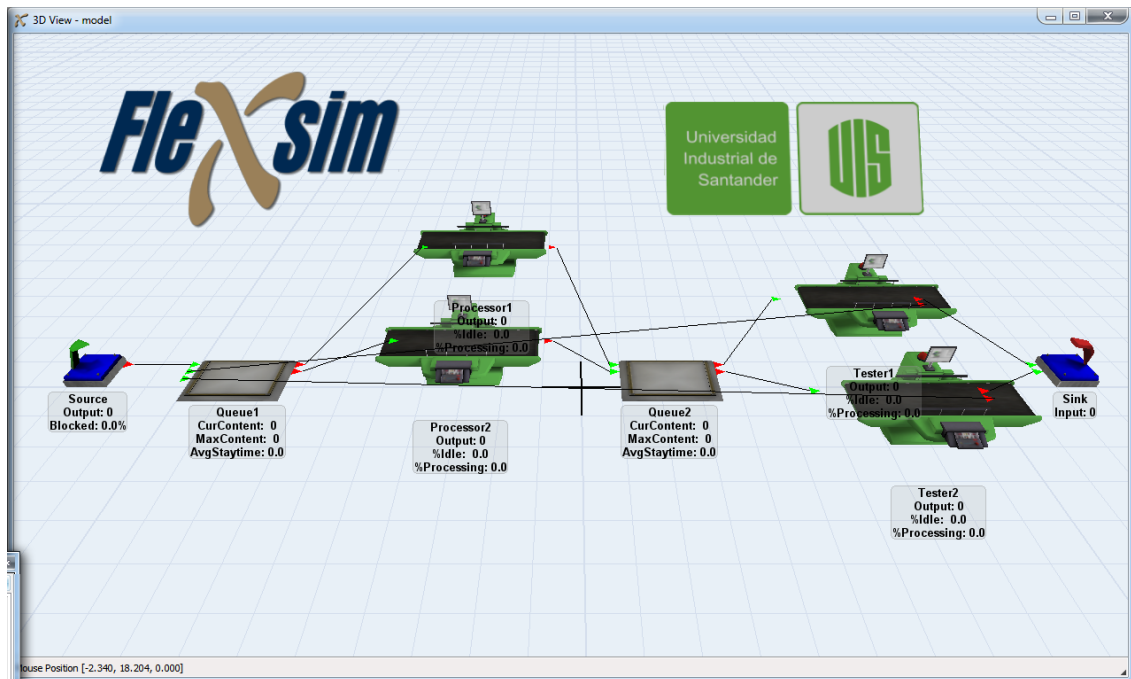


Es utilizado para registrar y/o mostrar información gráfica del modelo. Captura datos de tablas y datos definidos por el usuario.

## 2.5 VISTAS DEL MODELO

Flexsim utiliza un ambiente de modelado tridimensional. La vista del modelo por default para construir modelos se llama *3D View* o vista en perspectiva, que aparece cuando se crea un modelo nuevo o si se presiona el botón “3D” de la barra de herramientas. También se puede construir el modelo con la vista planar y utilizar la vista en perspectiva cuando se quiera visualizar o mostrar el modelo cuando se ejecuta la simulación. Se pueden abrir cuantas ventanas de vistas se deseen en Flexsim simultáneamente y cada una puede mostrar partes diferentes del modelo, aunque entre más ventanas de vistas se tengan abiertas mayor será la demanda de recursos del computador.

**Figura 35. Vista 3D del modelo (vista frontal o front)**



Fuente: Autor.

Figura 36. Vista 3D del modelo (superior o top)

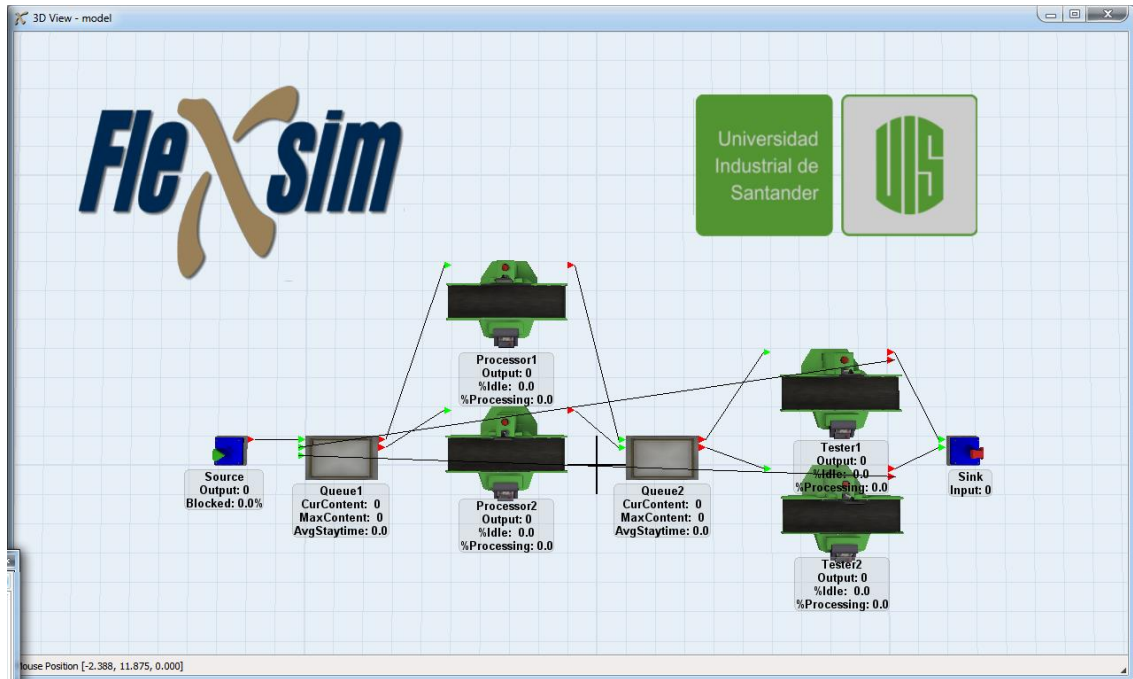
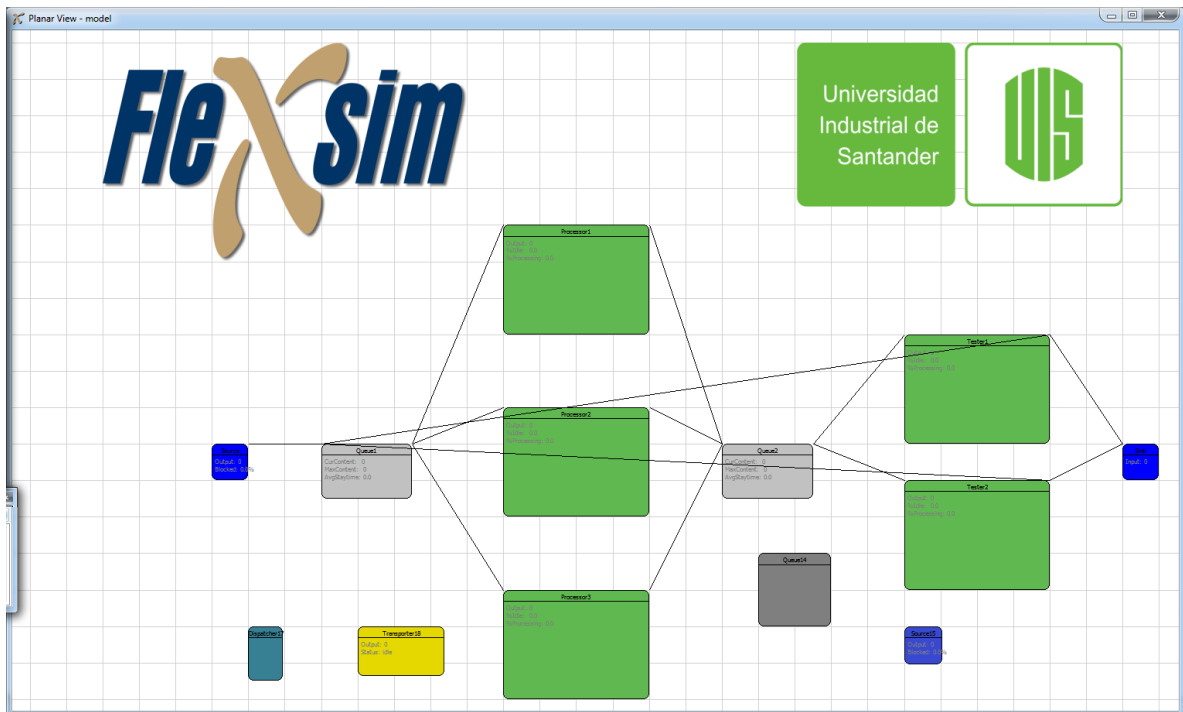
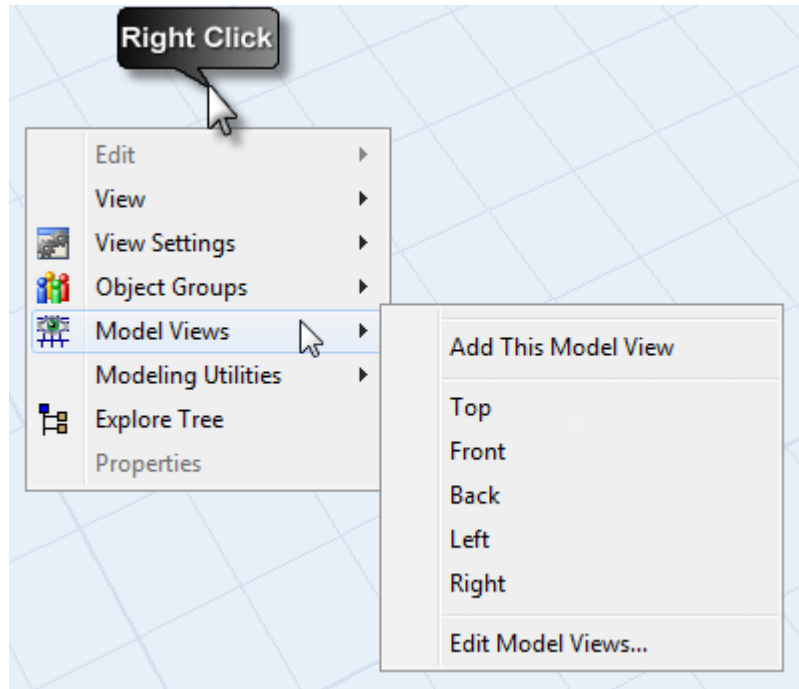


Figura 37. Vista Planar del modelo (diagrama en bloques)



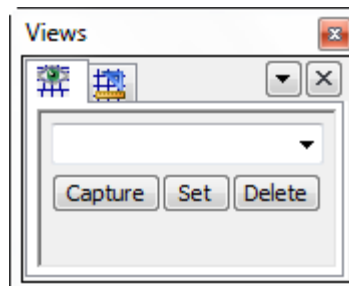
Para visualizar las vistas predeterminadas de Flexsim en la vista en perspectiva o 3D, se da clic derecho en un área vacía del layout y se selecciona la opción *Model Views* del menú contextual emergente, posteriormente se escoge una de las cinco (5) opciones que muestra la imagen a continuación.

**Figura 38. Model Views**



Se pueden agregar vistas personalizadas del layout o de áreas de interés del modelo, escogiendo la primera opción del submenú anterior, con lo cual aparecerá en una ventana la utilidad de selección de vistas del usuario disponibles.

**Figura 39. Utilidad Views**



Fuente: Interfaz de Flexsim.

### 3. CONSTRUCCIÓN DE MODELOS

#### 3.1 ELEMENTOS DE UN MODELO DE SISTEMA<sup>24</sup>

A continuación se mencionan los principales elementos de un modelo de simulación junto con su argot en Flexsim:

- ❖ **Entidades (Flowitems):** las entidades son los objetos que son creados para moverse a través del sistema. Pueden representar objetos reales o de conceptos más abstractos.
- ❖ **Recursos (Flexsim Object):** elementos u objetos que componen el sistema y que no son entidades.
- ❖ **Localizaciones (Fixed Resources):** lugares por donde deben esperar o fluir las entidades, son recursos fijos.
- ❖ **Atributos (Itemtypes):** características de una entidad.
- ❖ **Etiquetas (labels):** características o atributos de un objeto de Flexsim.
- ❖ **Variables (Variables, Labels):** condiciones cuyos valores se crean y modifican.
- ❖ **Controladores (Triggers, Flow Tab...):** reglas que gobiernan el sistema.

---

<sup>24</sup> MASINI, Javier. (2011). *Construcción de modelos usando Flexsim*. Recuperado el 10 de septiembre de 2012 en: <http://www.slideshare.net/jmasini/construccin-de-modelos-usando-flexsim>

- ❖ **Evento (Event):** cambio en el estado del sistema
  
- ❖ **Redes de transporte (Networks):** caminos o rutas por las que circulan los recursos móviles y las entidades de una estación o localización a otra.
  
- ❖ **Estados del sistema (States):** condición del sistema en un momento t.

Una vez identificados los elementos del sistema, se recomienda hacer la construcción del modelo en el siguiente orden:

1. Entidades
2. Localizaciones
3. Llegadas
4. Proceso
5. Variables, etiquetas, macros
6. Redes de transporte
7. Eventos de usuario
8. Experimenter
9. OptQuest

Para realizar una adecuada construcción del modelo, es necesario desarrollar previamente un modelo conceptual o cualitativo del sistema de interés, apoyándose en la documentación existente del sistema o haciendo el levantamiento de la información relevante a partir de datos históricos, diagramas de proceso, de flujo, etc.

### 3.2 TERMINOLOGÍA DEL SOFTWARE FLEXSIM

Todo software tiene su propia terminología que es conveniente conocer antes de comenzar a utilizarlo. A continuación se definen algunos términos de Flexsim.

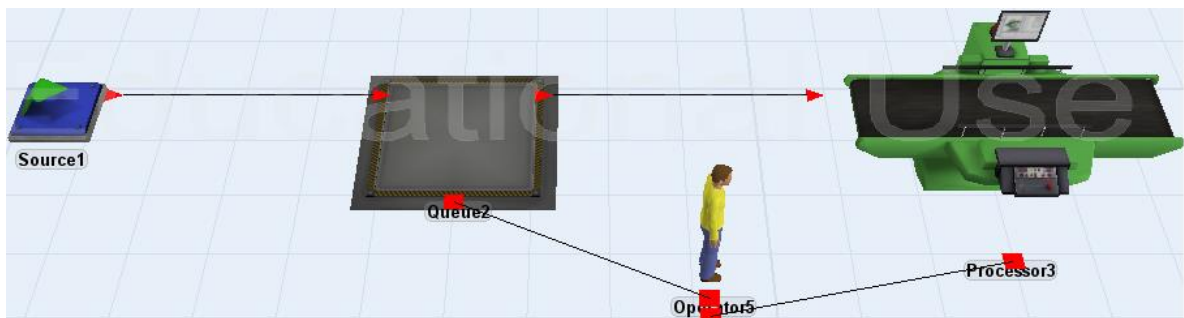
#### ❖ Flexsim Objects:

En Flexsim los objetos son el elemento de construcción de una simulación más básico. Estos objetos simulan diferentes tipos de recursos en la simulación. Como un ejemplo tenemos al objeto llamado Source, que genera los objetos o flowitems que circularán por el sistema, es decir, en este objeto se simulan las llegadas o entradas del sistema modelado.

Otro objeto es el llamado Queue, el cual actúa como un buffer o un área de almacenamiento. El Queue puede representar una fila de personas esperando, una fila de procesos que esperan ser procesados por una computadora, un área de almacenamiento en el piso de una fábrica, o bien una fila de llamadas en espera de un centro telefónico de servicio a clientes.

Otro ejemplo de un objeto de Flexsim es el objeto llamado Processor, que simula un tiempo de demora o de proceso. Este objeto puede representar una máquina de una fábrica, un cajero atendiendo a un cliente en un banco, etc.

**Figura 40. Objetos de Flexsim**



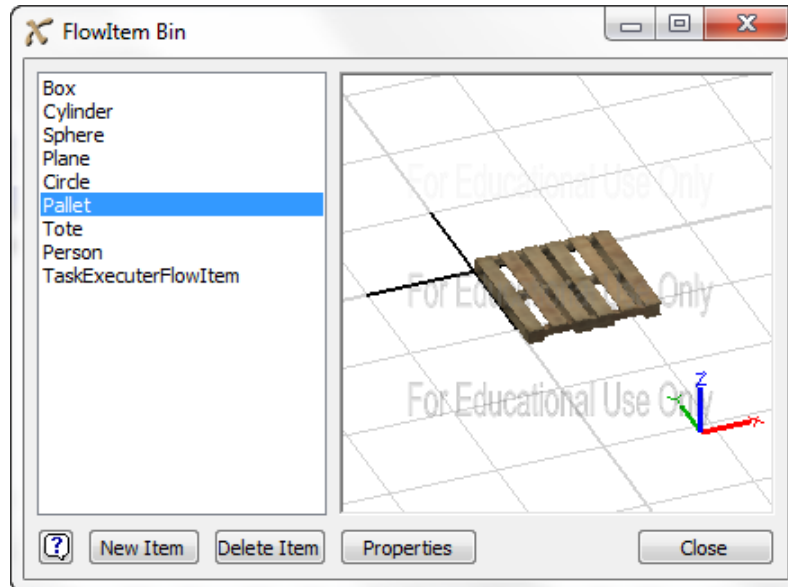
A todos los objetos de Flexsim se les puede modificar su apariencia fácilmente cambiando el dibujo 3D que tienen asignado. Los objetos de Flexsim se encuentran en la Biblioteca de Objetos. Esta biblioteca está ordenada por grupos. El grupo más utilizado es el que siempre se muestra primero. Entre los objetos que posee el software Flexsim se pueden mencionar el Source (fuente de entidades), Queue (cola), Sink (salida de entidades), Conveyor (correas transportadoras), Combiner (combinador), Separator (separador), Transporter (transporte), Operator (operador), etc.

❖ **Flowitems:**

Los ítems de flujo o entidades son los objetos que se mueven a través del modelo. Estos ítems pueden ser productos, partes, tarimas, ensamblajes, papeles, contenedores, llamadas telefónicas, papel, contenedores o cualquier cosa que se mueva a través del modelo. La mayoría tienen procesos que se realizan sobre ellos o son transportados a través del modelo por otros recursos.

En Flexsim, los flowitems son creados con el objeto denominado Source. Una vez que los flowitems han pasado a través del modelo, estos son enviados a un objeto llamado Sink, que se pone al final del proceso y destruye los flowitems. Los flowitems están listados en la herramienta *Flowitem Bin* accesible desde el menú Tools.

Figura 41. Contenedor de Flowitems



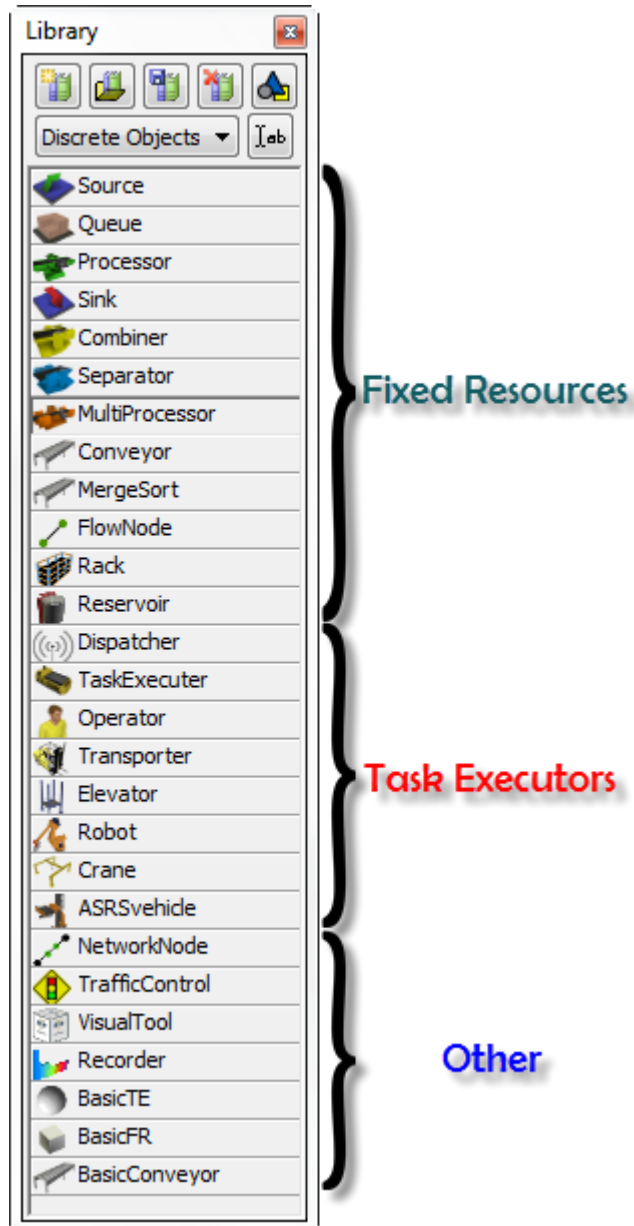
Fuente: Interfaz de Flexsim.

Figura 42. Ítems de Flujo o Flowitems



**Categorías de Objetos Discretos:** las simulaciones necesitan objetos que interactúen con los flowitems o entidades del sistema. Estos objetos pueden desarrollar una operación, crear una espera o mover los ítems. En Flexsim, hay dos categorías generales de objetos discretos: **Fixed Resources** y **Task Executors (Mobile Resources)**, es decir, recursos fijos y recursos móviles respectivamente.

Figura 43. Clases de Objetos Discretos



Fuente: Flexsim.

- ❖ **Fixed Resources:** son los objetos que envían, reciben y desarrollan actividades u operaciones en los flowitems o entidades. Son el tipo más común de objeto y se llaman recursos fijos porque son estacionarios.

❖ **Task Executors o Mobile Resources:** son recursos móviles que tienen la capacidad de moverse libremente dentro del modelo ejecutando las tareas que les han sido asignadas. Pueden transportar flowitems o ser usados como un recurso (operario o transporte) por los recursos fijos.

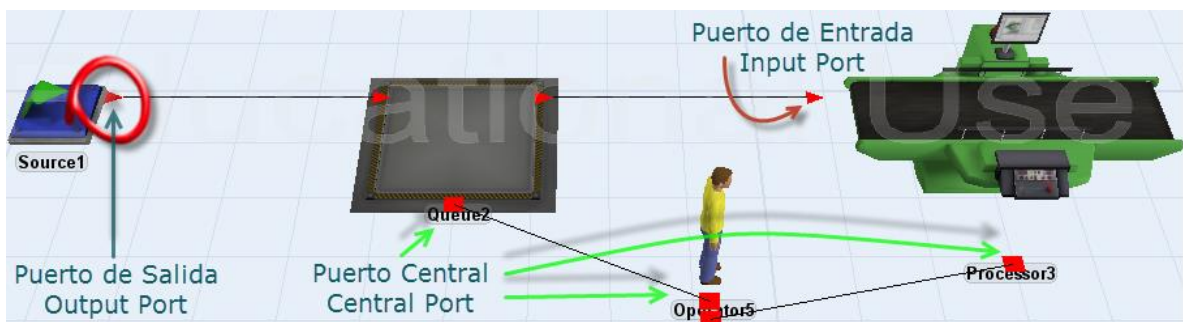
❖ **Itemtype:**

El tipo de ítem es una etiqueta que tienen todos los ítems de flujo (flowitem) o productos y puede representar un código de barras, un tipo de producto o el número de una pieza. Una entidad (Flowitem) define la clase básica de producto o pieza, por otra parte el tipo de ítem define el tipo de pieza individual o el número de pieza dentro de las clases de entidades. Flexsim está preparado para utilizar el Itemtype como una referencia para decidir la ruta o el destino al cual deben dirigirse los flowitems.

❖ **Ports:**

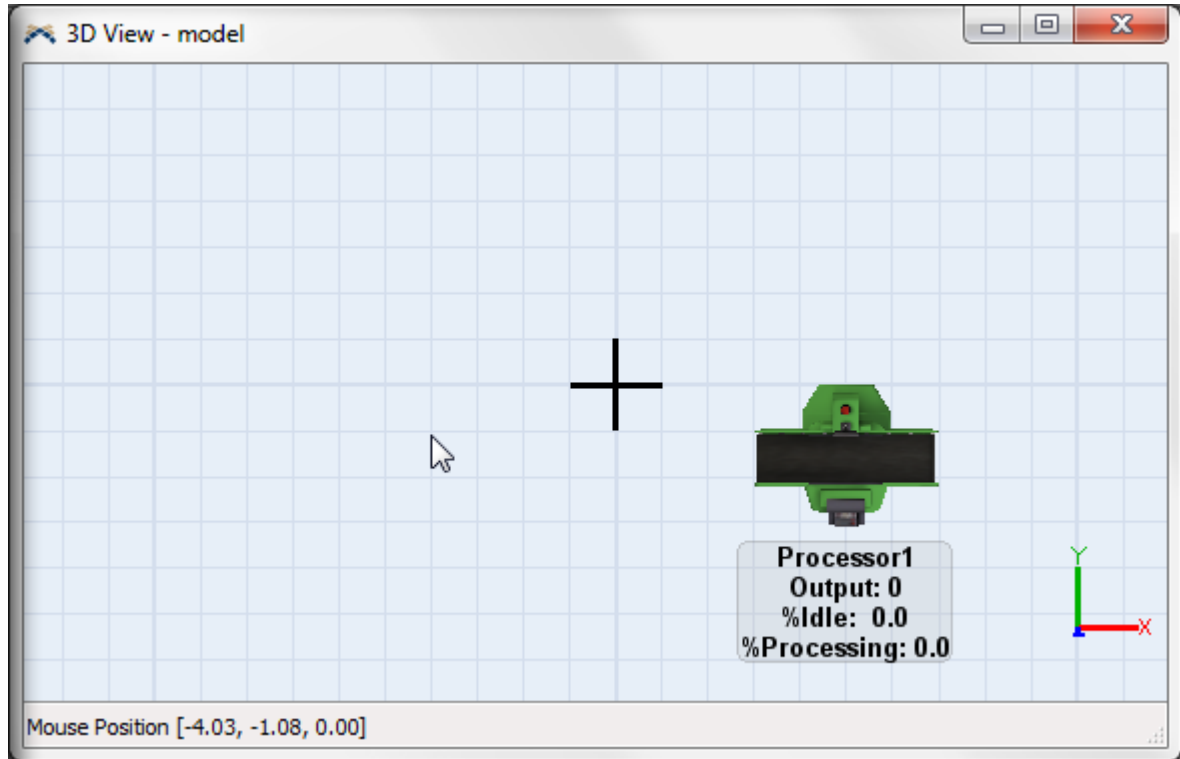
Todo objeto en Flexsim tiene un número ilimitado de puertos a través de los cuales se comunican con otros objetos. Existen tres tipos de puertos: puertos de entrada (input ports), puertos de salida (central ports) y puertos centrales (central ports). Los puertos de entrada y de salida se utilizan para el ruteo de los ítems de flujo, sea de manera automática o a través de recursos móviles. Los puertos centrales son usados para crear referencias, o punteros entre los objetos, los cuales son usados para conectarse a recursos de transporte.

**Figura 44. Puertos en Flexsim**



### 3.3 NAVEGACIÓN EN LA VISTA DE MODELO

Figura 45. Vista inicial del modelo

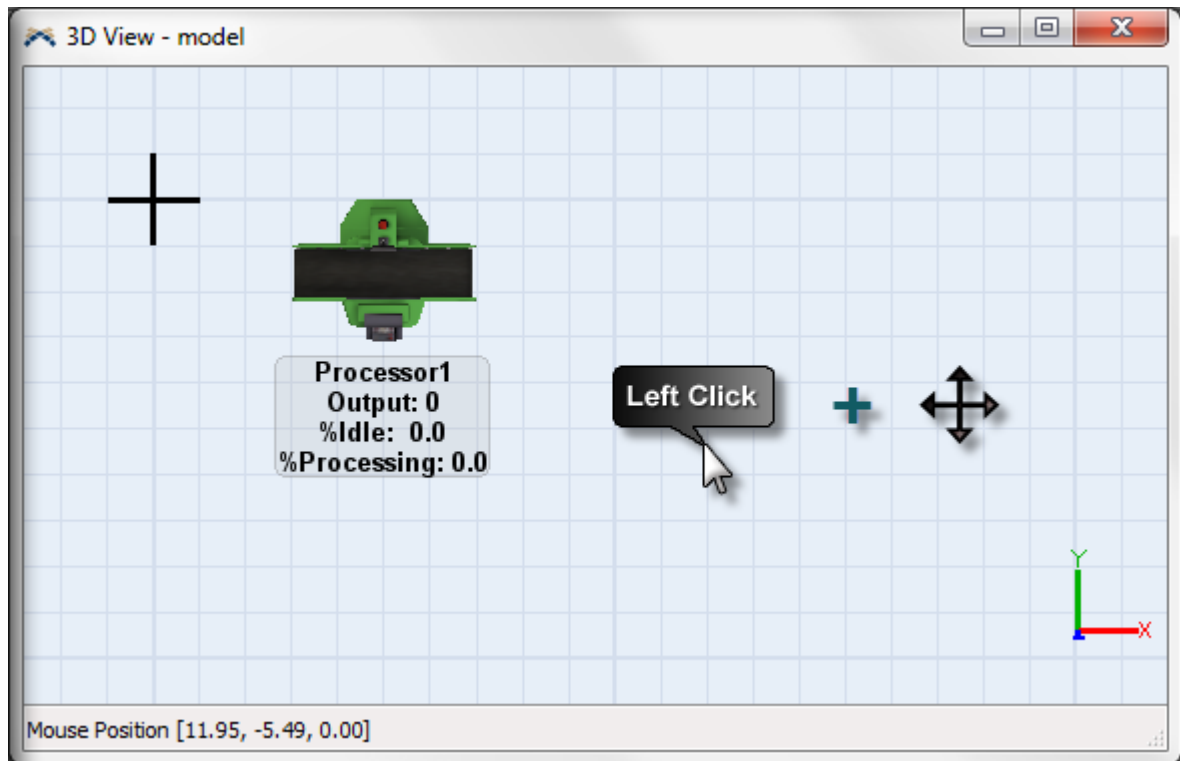


La vista de modelo puede desplazarse de diversas formas:

❖ **Con clic izquierdo sostenido:**

- ✓ Sobre el layout realiza el desplazamiento del modelo en el plano X-Y moviendo el mouse a la posición deseada.
- ✓ Si es sobre un objeto este se desplaza en el plano X-Y.

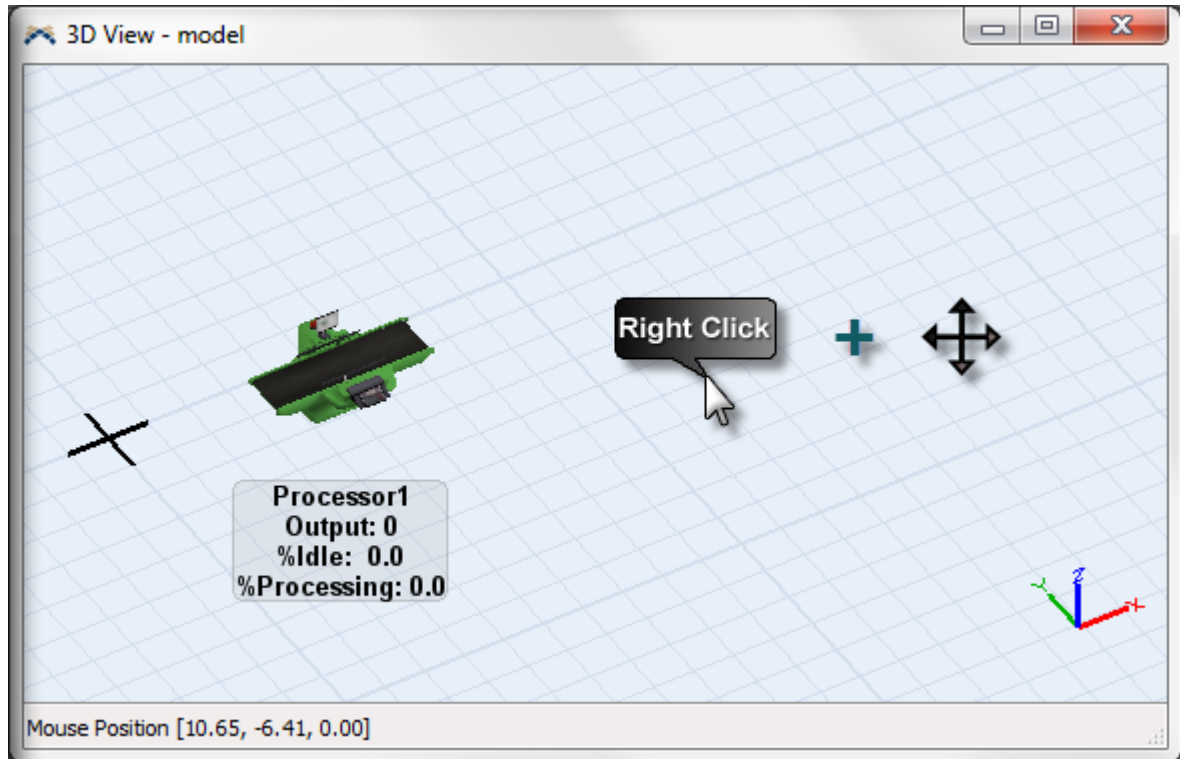
Figura 46. Desplazamiento del layout en el plano X-Y



❖ **Con clic derecho sostenido:**

- ✓ Sobre el layout realiza la rotación en X, Y, Z moviendo el mouse.
- ✓ Si es sobre un objeto, rota si el clic derecho es sobre uno de los ejes.

Figura 47. Rotación del layout en X-Y-Z



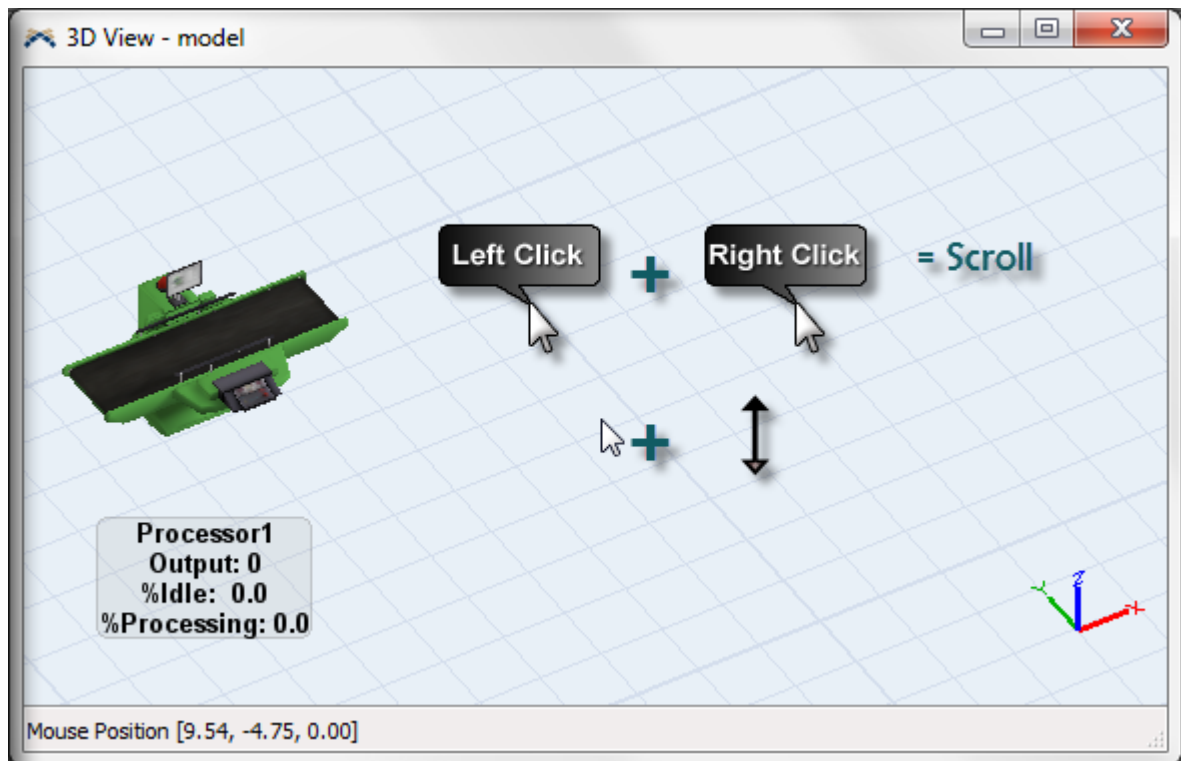
❖ **Con el scroll del mouse o botón izquierdo y derecho simultáneamente:**

- ✓ Sobre el layout hace zoom a la vista de modelo. Zoom adelante o hacia arriba acerca y zoom atrás o hacia abajo aleja.
- ✓ Sobre un objeto seleccionado (contorno amarillo alrededor de este) modifica su altura en el eje Z.

Figura 48. Botón scroll del mouse



Figura 49. Zoom de acercamiento



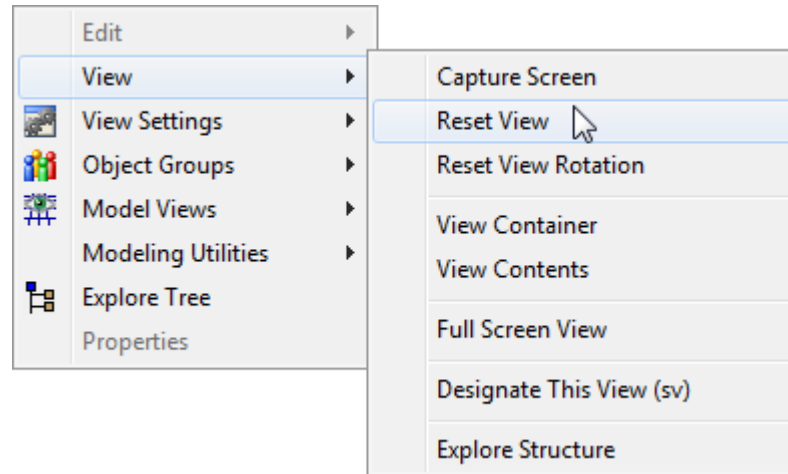
- ❖ **Tecla F8:** presionando la tecla F8 se activa el modo de vuelo donde se puede mover el cursor del mouse hacia arriba de la línea central de la ventana para volar hacia delante, debajo de la línea central para volar hacia atrás, a la izquierda de la línea central para rotar hacia la izquierda y hacia la derecha para rotar en esa otra dirección. Para salir del modo de vuelo simplemente se presiona nuevamente la tecla F8.

**Figura 50. Modo de vuelo**



Si en determinado momento el usuario se aleja y se pierde volando o desplazando el modelo, se puede reiniciar la vista con un solo clic con el botón derecho sobre el layout seleccionando la opción *View* y posteriormente *Reset View*.

Figura 51. Reinicio de la vista de modelo



### 3.4 CREACIÓN DE OBJETOS

Se pueden crear objetos de dos maneras:

1. Los objetos de Flexsim se insertan en el layout arrastrándolos con clic izquierdo sostenido desde el panel *Library* y arrastrando el objeto con el mouse (acción drag and drop) a la posición que se desee en cualquiera de las vistas del modelo, donde se representa el layout del sistema, por ejemplo la ventana *3D View - model*.

Figura 52. Creación de Objetos – Paso 1: Clic

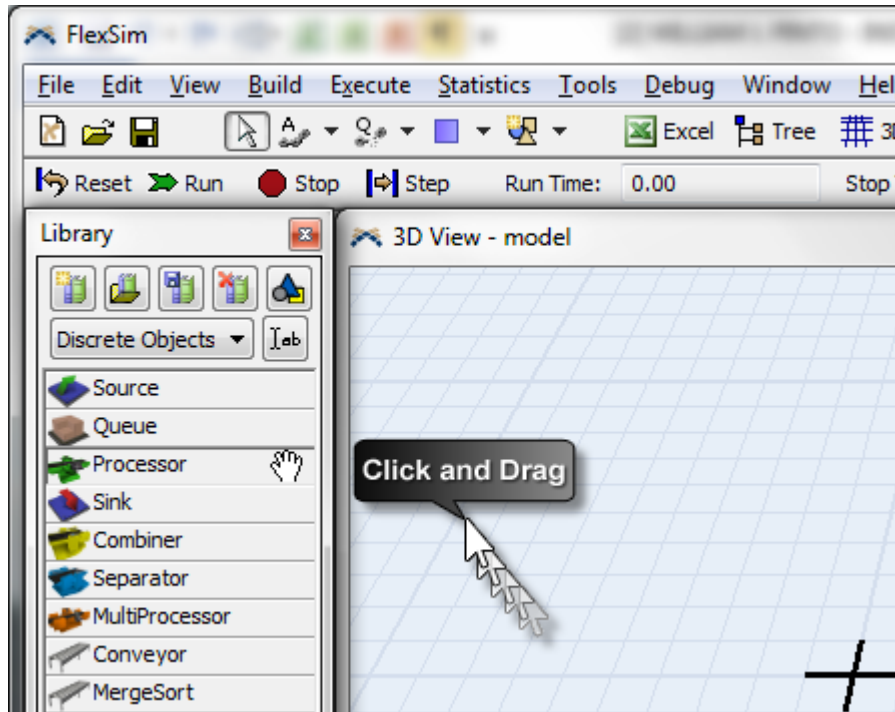
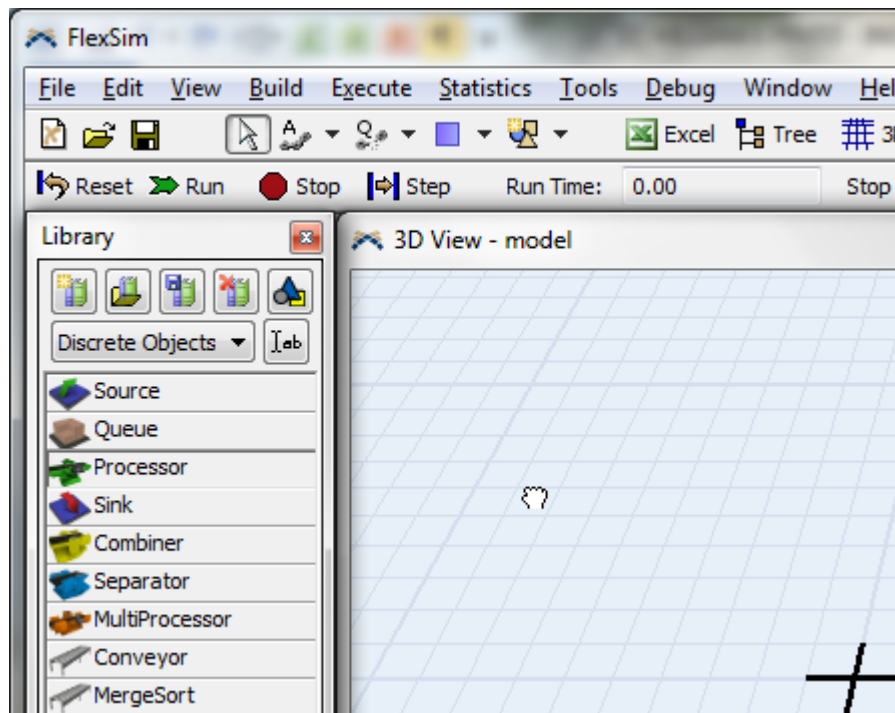
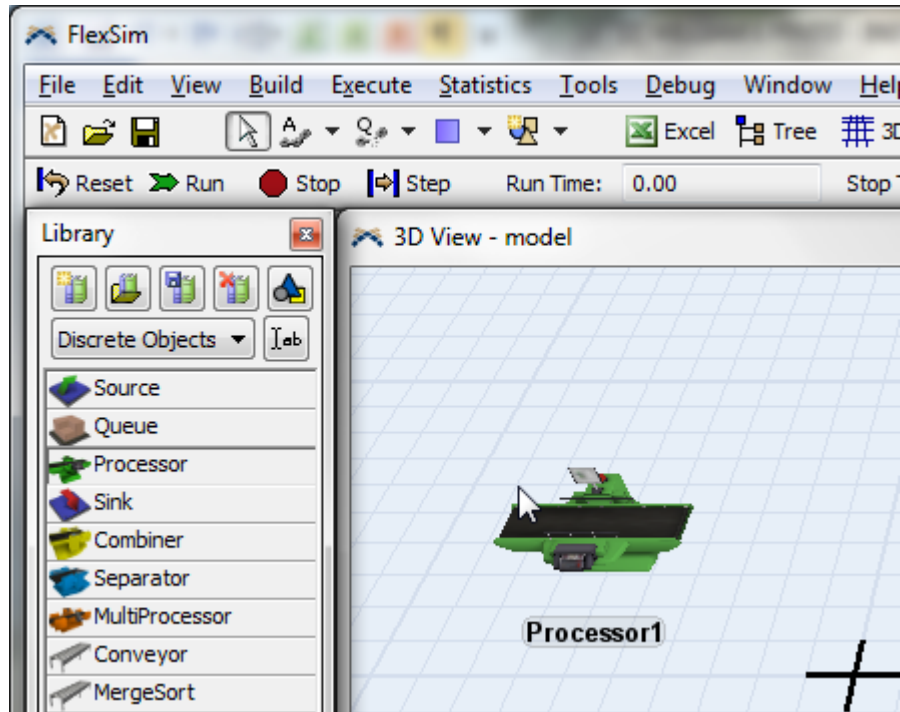


Figura 53. Creación de Objetos – Paso 2: Arrastrar

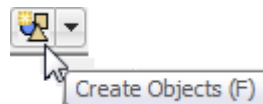


**Figura 54. Creación de Objetos – Paso 3: Soltar**



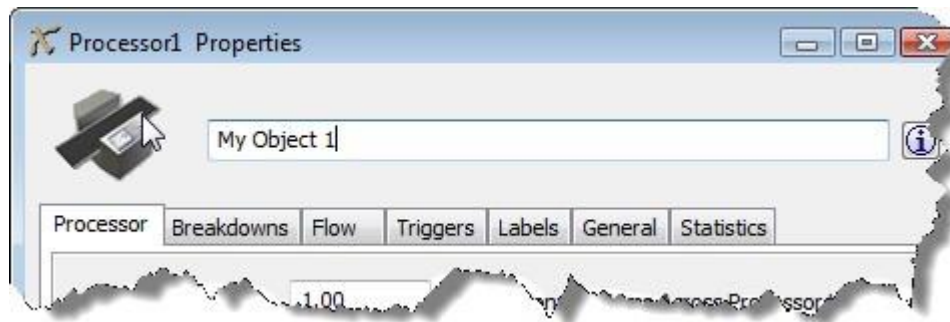
2. Se clikea en el botón de la barra de herramientas que se muestra en la figura 48, a continuación se da clic en el objeto de la librería que desea crear y por último clic en la vista del modelo donde desee agregar el objeto creado Si se da clic nuevamente en el layout agregará otro objeto similar.

**Figura 55. Creando objetos**



Una vez creada la entidad en el modelo, Flexsim asignará el nombre por defecto del objeto seguido de un numeral, por ejemplo *Processor#*, indicando el número de objeto creado desde que la aplicación fue abierta. Este nombre puede ser cambiado desde la ventana de propiedades, para lo cual se debe hacer doble clic en el objeto.

Figura 56. Renombrando objetos



No utilice caracteres especiales como >, <, @, {, (, \*, etc. al asignar nombres a los objetos, los únicos caracteres no alfanuméricos permitidos son los espacios y guiones bajos, tampoco comience con un carácter numérico. Es importante que cada objeto tenga un nombre único.

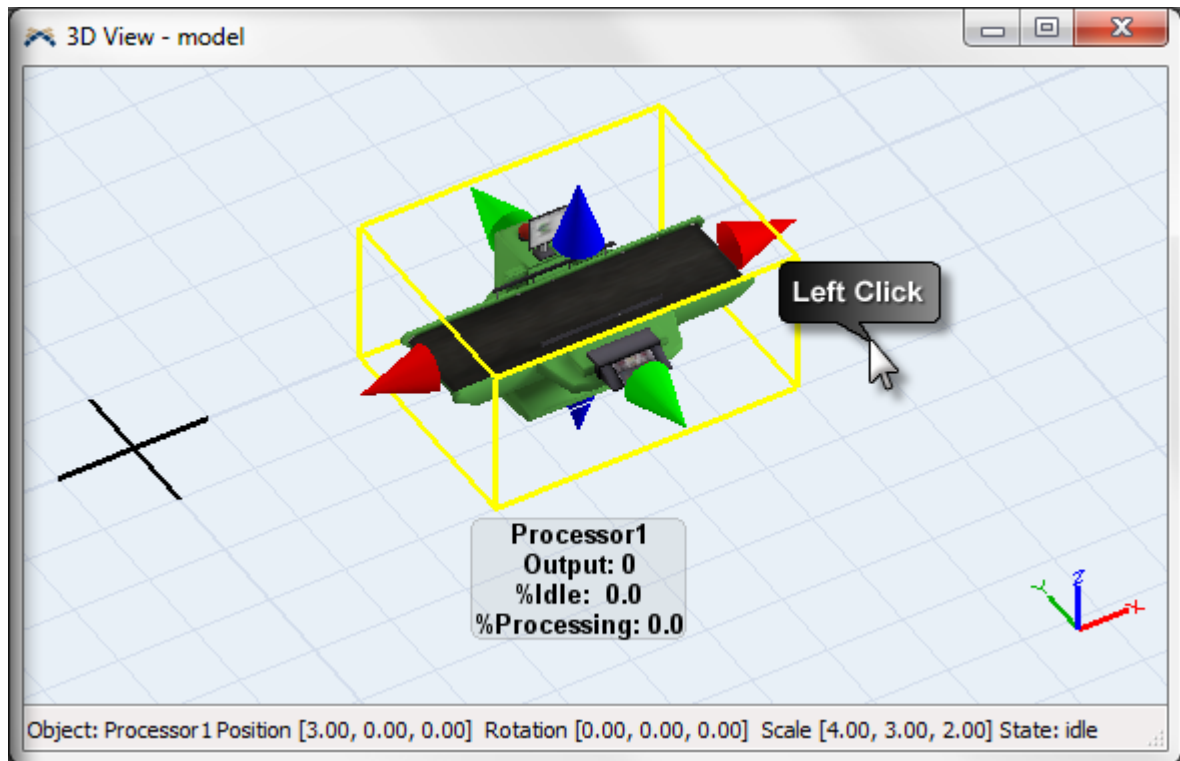
**RECOMENDACIÓN:** es aconsejable utilizar una convención sencilla y clara para nombrar los objetos de acuerdo a su tipo; por ejemplo pr\_Fresadora, qu\_FilaClientes, cv\_BandaAlimentos, nn\_Inspeccion, indican que el objeto asociado es un procesador (Processor), una cola (Queue), una banda transportadora (Conveyor) y un nodo (Network Node) respectivamente.

### 3.5 EDICIÓN DE OBJETOS

- ❖ **Seleccionando objetos:** basta con hacer clic izquierdo sobre este para que aparezca un contorno amarillo alrededor del objeto junto con sus ejes en forma de conos en sus extremos. Los ejes coordenados y sus colores asociados son:

<b>Rojo:</b>	Eje <b>X</b> .
<b>Verde:</b>	Eje <b>Y</b> .
<b>Azul:</b>	Eje <b>Z</b> .

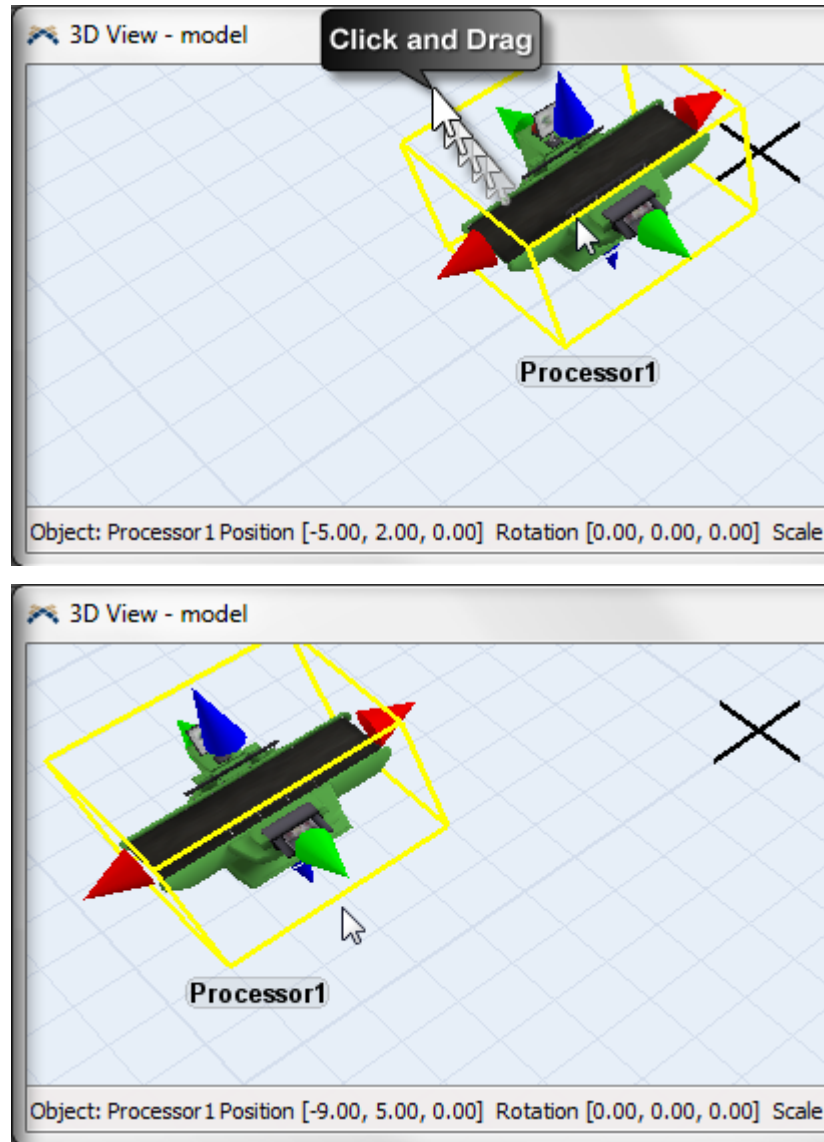
Figura 57. Seleccionando objetos



- ❖ **Movimiento de objetos:** para mover un objeto basta con hacer clic izquierdo y arrastrarlo a su nueva posición. En la barra de estado (parte inferior de la ventana de vista del modelo) aparece indicado el nombre, la posición, rotación, escala y estado del objeto seleccionado.

Al ubicarlo en una nueva posición se puede verificar (no solo visualmente) en esta barra de estado las coordenadas correspondientes a la nueva posición.

Figura 58. Moviendo objetos



Como se observa en la imagen inferior de la figura 58, las coordenadas del objeto cambian respecto de la imagen superior.

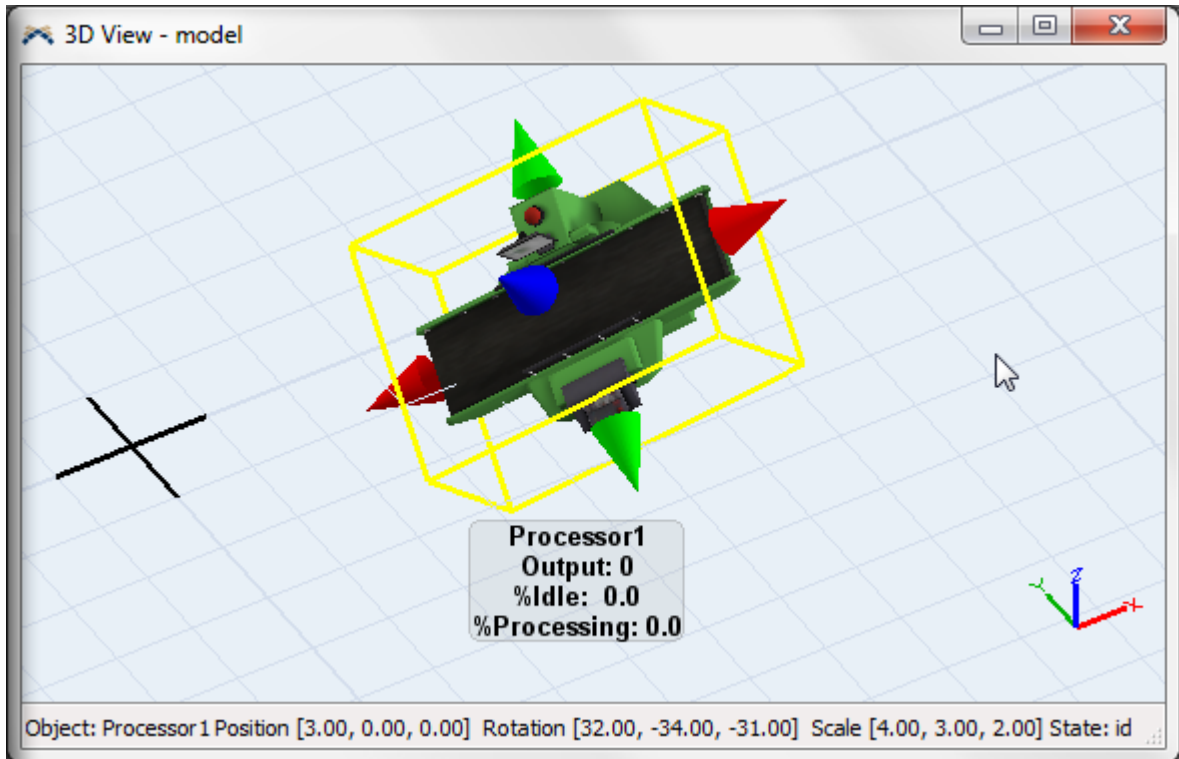
- ❖ **Acercando o alejando objetos:** para desplazar un objeto en el eje Z, se selecciona con clic y se hace scroll (girar la rueda del mouse). En la barra de estado aparecerá la nueva coordenada de Z.

**Figura 59. Desplazamiento en el eje Z**



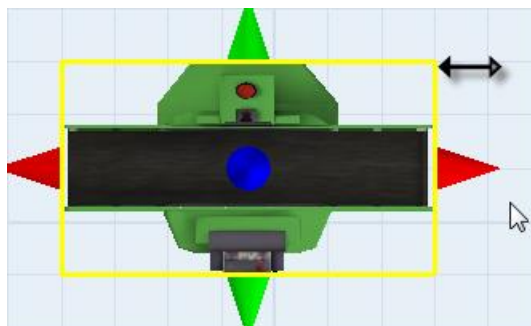
- ❖ **Rotando objetos:** para editar la rotación de un objeto se hace clic derecho sostenido sobre la flecha correspondiente al eje que se desea girar alrededor y se arrastra el mouse hacia adelante o hacia atrás.

Figura 60. Rotación de objetos



- ❖ **Destruyendo objetos:** selecciónelo y oprima la tecla *Del* o *Suprimir*.
- ❖ **Redimensionamiento:** para cambiar la dimensión de un objeto, primero selecciónelo y luego de clic sostenido en el cono del eje que desee cambiar, arrastrando el mouse hacia arriba o abajo. En la imagen a continuación, se redimensiona en X.

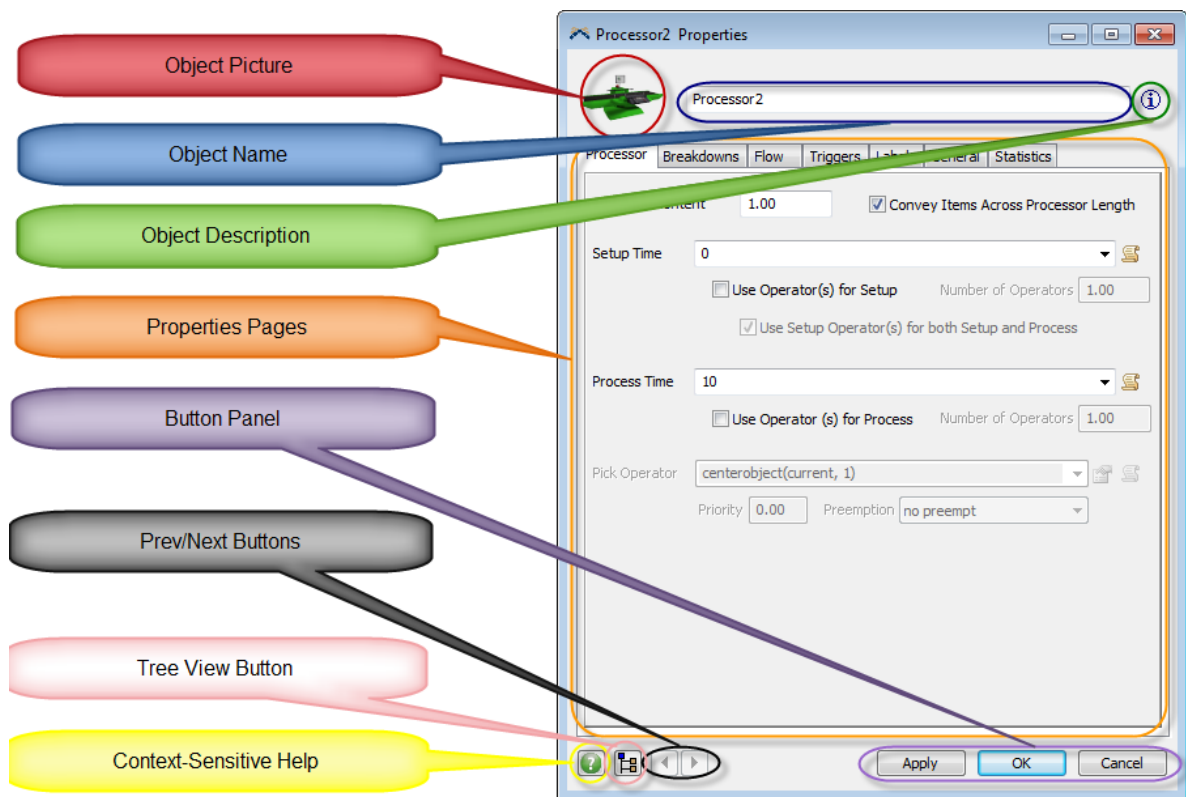
Figura 61. Rotación de objetos



### 3.6 VENTANA DE PROPIEDADES

Todos los objetos en Flexsim tienen un número de pestañas que presentan información que el modelador puede cambiar basado en requerimientos del modelo. La ventana de propiedades mostrada en la figura 61, es la vista por defecto que tiene un objeto cuando es seleccionado con doble clic.

**Figura 62. Propiedades del objeto**



Fuente: Ayuda de Flexsim.

- ❖ Object Picture: muestra la imagen por defecto del tipo de objeto que está editando.
- ❖ Object Description: este botón abre un campo de texto para describir y documentar el objeto.

- ❖ Properties Pages: conjunto de pestañas de propiedades de los objetos. Pueden variar dependiendo del objeto en cuestión, sin embargo, hay una serie de pestañas compartidas por múltiples objetos:

**Tabla 3. Pestañas compartidas de propiedades**

PESTAÑAS COMPARTIDAS	
PESTAÑA	OBJETOS
Breakdowns	Processor, Combiner, Separator
Breaks	TaskExecuter, Transporter, Operator, Crane, ASRSvehicle, Robot, Elevator
Collisions	TaskExecuter, Transporter, Operator, Crane, ASRSvehicle, Robot, Elevator
Conveyor	Conveyor, MergeSort, BasicConveyor
Flow	Source, Queue, Processor, Combiner, Separator, MultiProcessor, Conveyor, Rack, Reservoir, FlowNode, MergeSort, BasicConveyor
General	Todos los objetos
Labels	Todos los objetos
Layout	Conveyor, MergeSort, BasicConveyor
ProcessTimes	Combiner, Separator
Statistics	Todos los objetos
TaskExecuter	TaskExecuter, Transporter, Operator, Crane, ASRSvehicle, Robot, Elevator
Triggers	Todos los objetos

Fuente: Flexsim.

**Tabla 4. Pestañas únicas de propiedades**

PESTAÑAS ÚNICAS DE PROPIEDADES			
PESTAÑA	OBJETO	PESTAÑA	OBJETO
ASRSvehicle		Photo Eyes	Conveyor
BasicFR Advanced		Processor	
BasicTE		Queue	
Combiner		Rack	
Container Functionality	Visual Tool	Recorder	
Crane		Reservoir	
Decision Points	BasicConveyor	Robot	
Dispatcher		Robot Geometry	
Display	Visual Tool	Separator	
Elevator		Sink	
Flowitem General	FlowItem Bin	SizeTable	Rack
FlowNode		Source	
MergeSort		Speeds	TrafficControl
MultiProcessor		Traffic Control	
NetworkNode		Transporter	

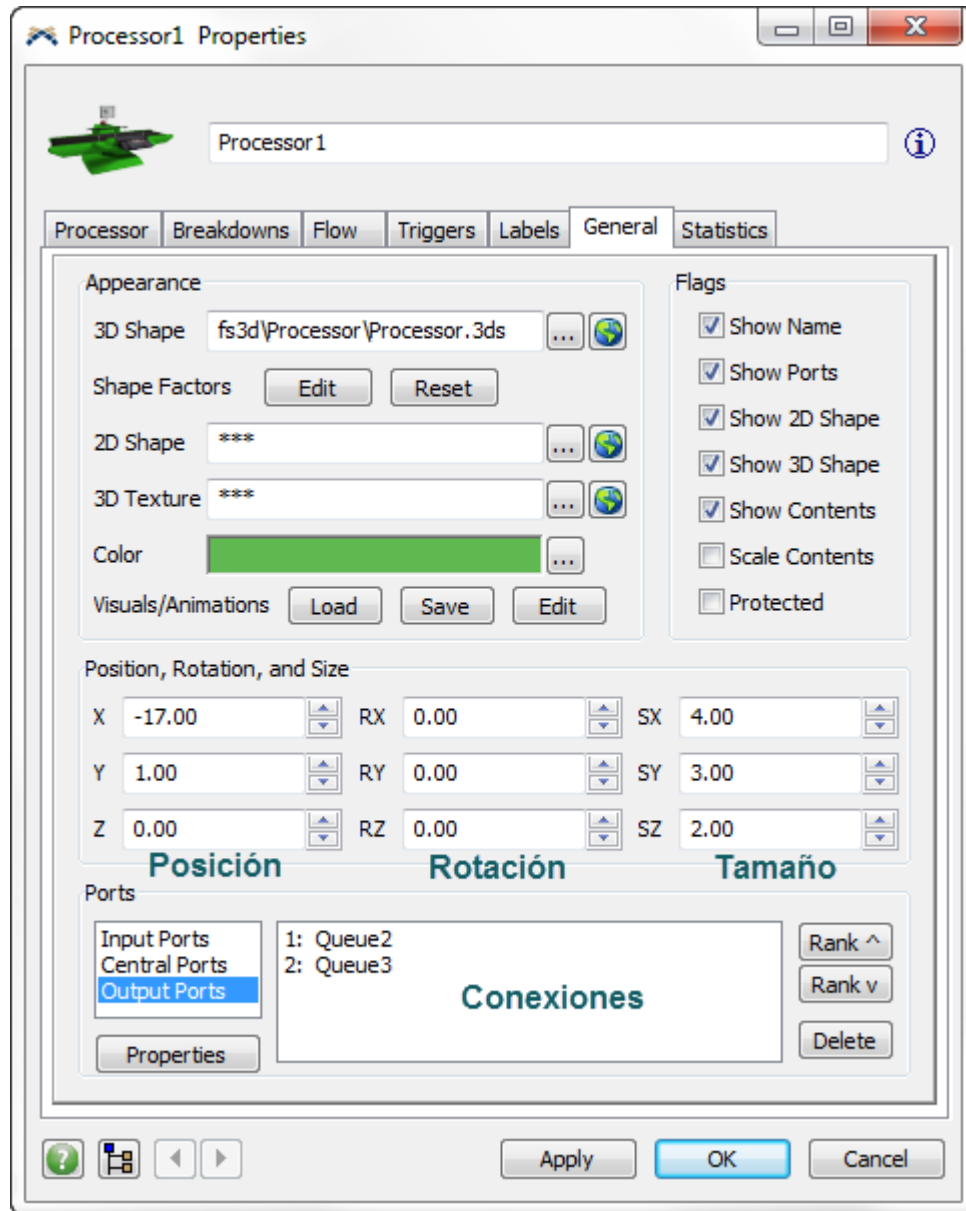
Fuente: Flexsim.

Las tablas 4 y 5 aplican para objetos discretos. Los objetos de fluidos también tienen pestañas que comparten la misma información.

Las pestañas en común contienen la siguiente información:

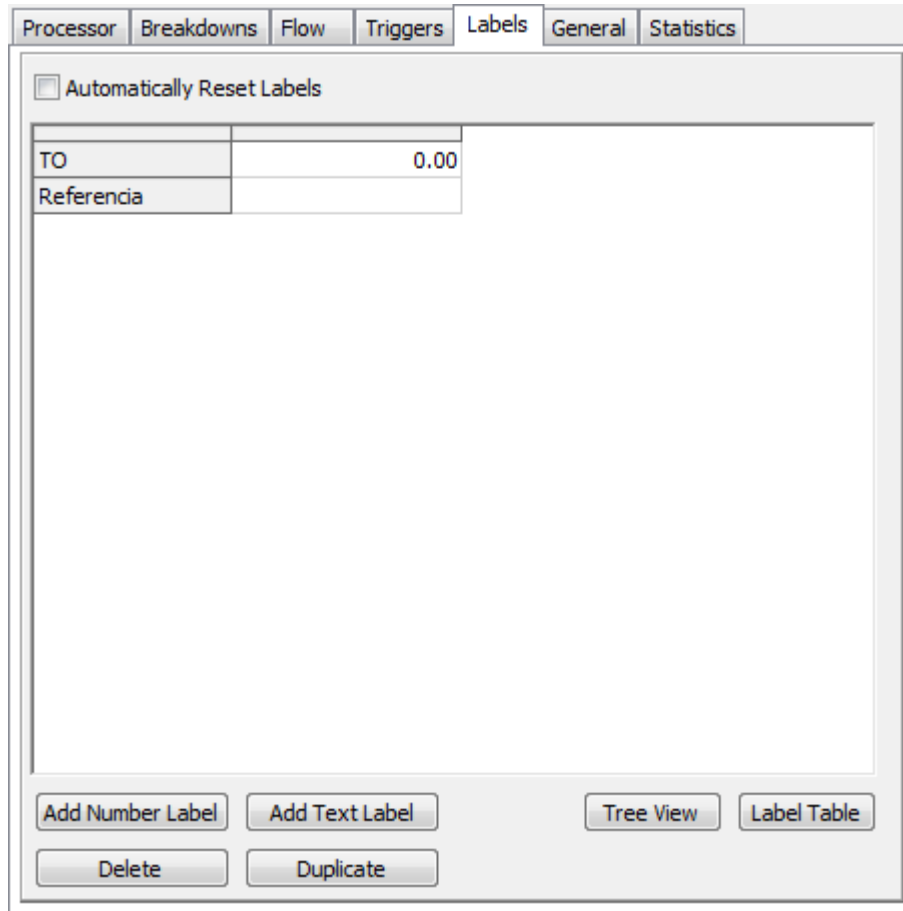
- ❖ **General:** atributos que afectan el aspecto visual, tamaño, posición y rotación de un objeto. También muestra la lista de conexiones de puertos del objeto.

**Figura 63. Pestaña General**



- ❖ **Labels:** se utiliza para crear etiquetas o atributos asociados al objeto, como una tabla de datos definida por el usuario.

**Figura 64. Pestaña Labels**



En la figura 64 se observa la ventana correspondiente a la pestaña *Labels*, que tiene dos etiquetas creadas: “TO” de tipo numérica, y “Referencia” de tipo texto.

- ❖ **Statistics:** contiene información estadística relativa al objeto, que es recopilada durante la corrida de simulación. Incluye tablas y gráficos.

Figura 65. Pestaña Statistics

The screenshot displays the 'Statistics' tab with the following sections and values:

- Throughput:** Input: 0.00, Output: 0.00, Total Travel Distance: (empty field)
- State:** Current: idle, Chart... button
- Record data for Content and Staytime charts
- Content:** Current: 0.00, Minimum: 0.00, Maximum: 0.00, Average: 0.00, Content History Size: 100.00, Chart... button
- Staytime:** Minimum: 0.00, Maximum: 0.00, Average: 0.00, Lower Bound: 0.00, Upper Bound: 100.00, Divisions: 20.00,  Display Confidence, Interval %: 95.00, Chart... button

- ❖ **Flow:** lógica de flujo, para determinar cómo se mueven las entidades o flowitems dentro y fuera del objeto.

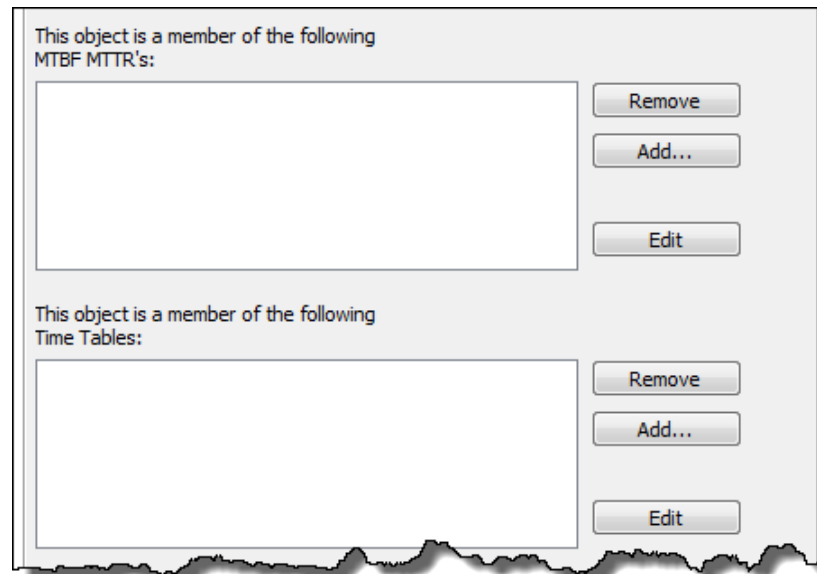
Figura 66. Pestaña Flow

The screenshot displays the 'Flow' tab with the following sections and values:

- Output:** Send To Port: First available, Use Transport:  centerobject(current, 1), Priority: 0.00, Preemption: no preempt,  Reevaluate Sendto on Downstream Availability
- Input:**  Pull Strategy: Any Port, Pull Requirement: Pull Anything

- ❖ **Breakdowns:** información acerca de la configuración de tiempos de falla y de paro programado.

**Figura 67. Pestaña Breakdowns**

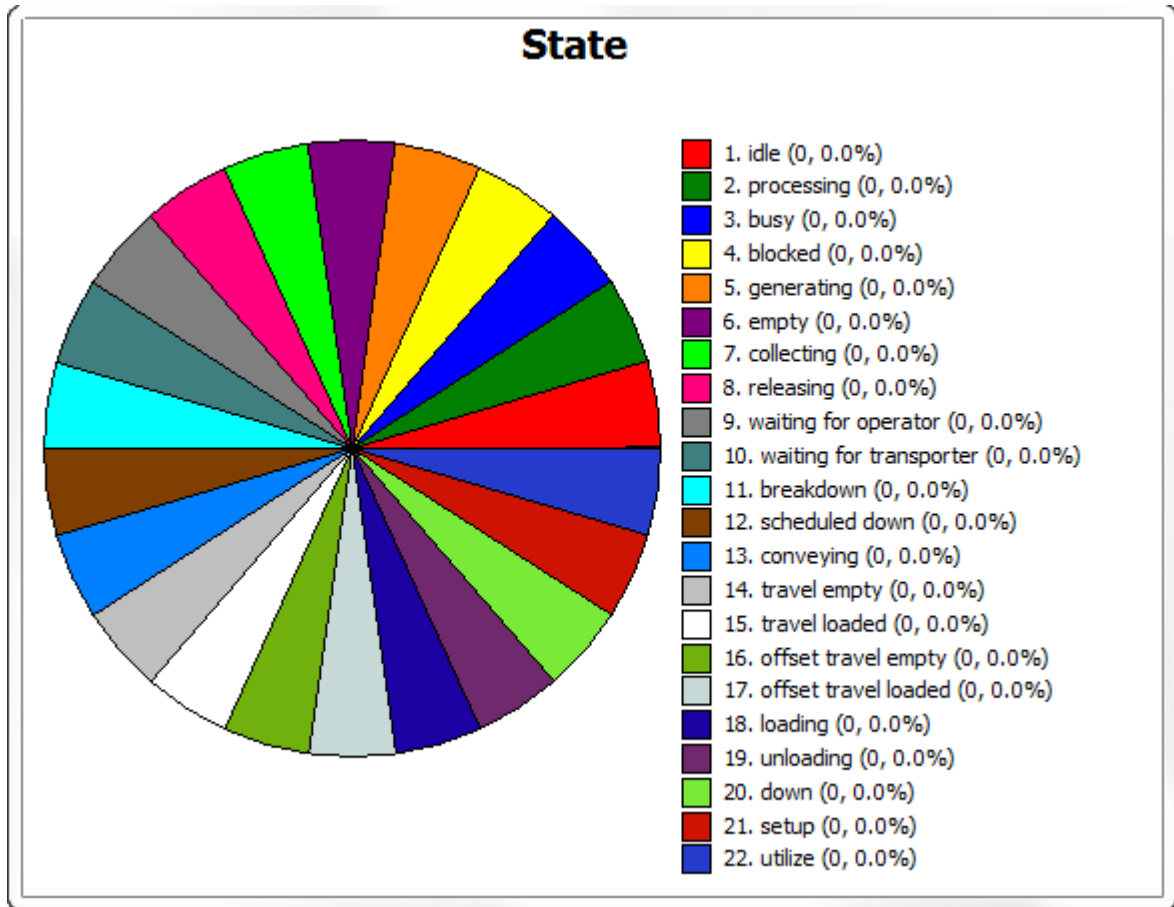


Fuente: Interfaz de Flexsim.

### 3.7 LISTA DE ESTADOS DE LOS OBJETOS

El estado de un objeto depende de su tipo, Flexsim tiene un total de 50 posibles estados, de los cuales algunos se presentarán durante la corrida de una simulación. Para acceder a los estados del objeto, es necesario entrar a la pestaña *Statistics* de la ventana propiedades, y a continuación presionar el botón *Chart*. Si el modelo está detenido, aparecerá una ventana con todos los estados disponibles para ese objeto. Si se encuentra en ejecución, visualizará los estados que apliquen en ese momento.

Figura 68. Principales estados de los objetos discretos en Flexsim



Fuente: Interfaz de Flexsim.

En la tabla 6 se listan todos los estados posibles de los objetos en Flexsim. Cada estado tiene asociado un valor numérico y su respectiva macro que es el texto que describe al estado.

A continuación se describen los estados que admiten los principales objetos:

**Tabla 5. Lista de estados de los objetos en Flexsim**

NÚM	ESTADO	NÚM	ESTADO
1	STATE_IDLE	26	STATE_STARVED
2	STATE_PROCESSING	27	STATE_MIXING
3	STATE_BUSY	28	STATE_FLOWING
4	STATE_BLOCKED	29	STATE_ALLOCATED_IDLE
5	STATE_GENERATING	30	STATE_OFF_SHIFT
6	STATE_EMPTY	31	STATE_CHANGE_OVER
7	STATE_COLLECTING	32	STATE_REPAIR
8	STATE_RELEASING	33	STATE_MAINTENANCE
9	STATE_WAITING_FOR_OPERATOR	34	STATE_LUNCH
10	STATE_WAITING_FOR_TRANSPORT	35	STATE_ON_BREAK
11	STATE_BREAKDOWN	36	STATE_SUSPEND
12	STATE_SCHEDULED_DOWN	37	STATE_AVAILABLE
13	STATE_CONVEYING	38	STATE_PREPROCESSING
14	STATE_TRAVEL_EMPTY	39	STATE_POSTPROCESSING
15	STATE_TRAVEL_LOADED	40	STATE_INSPECTING
16	STATE_OFFSET_TRAVEL_EMPTY	41	STATE_OPERATING
17	STATE_OFFSET_TRAVEL_LOADED	42	STATE_STANDBY
18	STATE_LOADING	43	STATE_PURGING
19	STATE_UNLOADING	44	STATE_CLEANING
20	STATE_DOWN	45	STATE_ACCELERATING
21	STATE_SETUP	46	STATE_MAXSPEED
22	STATE_UTILIZE	47	STATE_DECELERATING
23	STATE_FULL	48	STATE_STOPPED
24	STATE_NOT_EMPTY	49	STATE_WAITING
25	STATE_FILLING	50	STATE_ACCUMULATING

Fuente: Flexsim.

## Tipo de objeto: FixedResources

### ❖ Source o Fuente:

- **Generating:** No hay flowitems en el Source. Se está esperando hasta su próxima creación.
- **Blocked:** Los flowitems han sido creados y están esperando para dejar el Source.

### ❖ Queue o Fila/Cola:

- **Empty:** El Queue está vacío, es decir, sin flowitems.
- **Collecting:** El Queue está acumulando los flowitems de un lote.
- **Releasing:** El Queue ha finalizado de acumular los flowitems de su lote y los está liberando. Además, si el Queue no está acumulando o loteando y tiene flowitems, estará en este estado.
- **Waiting for Transport:** La cola tiene flowitems que han sido liberados y están listos para moverse aguas abajo, sino que se espera de un transporte para venir a recogerlos.

### ❖ Processor o Procesador:

- **Idle:** El objeto está vacío, es decir, sin flowitems.
- **Setup:** El objeto se encuentra en proceso de set up definido por el modelador.
- **Processing:** El objeto está en el tiempo de proceso definido por el modelador.
- **Blocked:** El(los) flowitem(s) ha(n) sido liberado(s), pero la(s) siguiente(s) estación(es) no está(n) lista(s) para recibirlo.
- **Waiting for Operator:** El objeto está esperando a que llegue un operador, ya sea para reparar una descompostura, o para operar un lote o batch.

- **Waiting for Transport:** El Processor ha liberado un flowitem, y el objeto que le sigue en el proceso está listo para recibirlo, pero aún tiene que esperar a un transporte que lo recoja.
  - **Down:** El Processor está descompuesto.
- ❖ Conveyer o Banda transportadora:
- **Empty:** No hay flowitems en el Conveyer.
  - **Conveying:** Todos los objetos están transportándose a lo largo del conveyer.
  - **Blocked:** El flowitem que va al frente ha llegado al final, pero la siguiente estación no está lista para recibirlo.
  - **Waiting for Transport:** El flowitem ha llegado al final del Conveyer pero está esperando a que el transporte llegue a recogerlo.
- ❖ Combiner o Combinador:
- **Idle:** El objeto no ha recibido ningún flowitem por el puerto 1.
  - **Collecting:** El objeto ha recibido un flowitem por el puerto 1, pero aún está recolectando flowitems por algún otro puerto.
  - **Setup:** El objeto está en proceso de set up definido por el modelador.
  - **Processing:** El objeto está en el tiempo de proceso definido por el modelador.
  - **Blocked:** El flowitem ha sido liberado, pero la siguiente estación no está preparada para recibirlo.
  - **Waiting for Operator:** El objeto está esperando a que llegue un operador, ya sea para reparar una descompostura, o para operar un lote o batch.
  - **Waiting for Transport:** El Combiner ha liberado un flowitem, y el objeto que le sigue en el proceso está listo para recibirlo, pero aún tiene que esperar a un transporte que lo recoja.
  - **Down:** El Combiner está descompuesto.

- ❖ Separator o Separador:
  - **Idle:** No hay flowitems en el Separator.
  - **Setup:** El objeto está en proceso de set up definido por el modelador.
  - **Processing:** El objeto está en el tiempo de proceso definido por el modelador.
  - **Blocked:** El flowitem ha sido liberado, pero la siguiente estación aún no está preparada para recibirlo.
  - **Waiting for Operator:** El objeto está esperando a que llegue un operador, ya sea para reparar una descompostura, o para operar un lote o batch.
  - **Waiting for Transport:** El Separator ha liberado un flowitem, y el objeto que le sigue en el proceso está listo para recibirlo, pero aún tiene que esperar a un transporte que lo recoja.
  - **Down:** El objeto está descompuesto.
  
- ❖ Rack o Almacén:
  - No involucra estados. Se usan las estadísticas de contenido (gráfico o numérico).

### Tipo de objeto: TaskExecuters

- ❖ Operator u Operador:
  - **Travel Empty:** El objeto está viajando hacia otro destino (objeto), pero no contiene ningún flowitem. Este estado se asocia exclusivamente a la tarea TASKTYPE\_TRAVEL.
  - **Travel Loaded:** El objeto se dirige hacia otro objeto cargando uno o más flowitems. Este estado se asocia exclusivamente a la tarea TASKTYPE\_TRAVEL.

- **Offset Travel Empty:** El objeto se comporta como offset travel y no contiene ningún flowitem.
- **Offset Travel Loaded:** El objeto se comporta como offset travel y carga uno o más flowitems.
- **Loading:** El objeto está tomando un flowitem. Este estado corresponde a la tarea TASKTYPE\_LOAD, y se aplica solamente al momento en que el objeto ha terminado su offset travel y espera su tiempo de carga definido por el modelador antes de cargar el elemento.
- **Unloading:** El objeto está descargando un flowitem. Este estado corresponde a la tarea TASKTYPE\_UNLOAD, y se aplica solamente al momento en que el objeto ha terminado su offset travel y espera su tiempo de descargue definido por el modelador antes de descargar el elemento.
- **Utilize:** El objeto está siendo utilizado por una estación. Este estado es usado por un operador, cuando llega a una estación y se le requiere para un proceso, set up o alguna reparación. Este estado es usualmente asociado con una tarea TASKTYPE\_UTILIZE, pero esta tarea también puede especificar un estado diferente. Además, otros tipos de tareas, como TASKTYPE\_DELAY, pueden utilizar el estado STATE\_UTILIZE.
- **Blocked:** El objeto que está viajando por alguna red es bloqueado por la misma red y no puede avanzar.

❖ Transporter o Transportador:

- **Travel Empty:** El objeto está viajando hacia otro objeto, pero no contiene ningún flowitem.
- **Travel Loaded:** El objeto se dirige hacia otro objeto cargando uno o más flowitems.
- **Offset Travel Empty:** El objeto se comporta como offset travel y no contiene ningún flowitem.

- **Offset Travel Loaded:** El objeto se comporta como offset travel y carga uno o más flowitems.
- **Loading:** El objeto está tomando un flowitem.
- **Unloading:** El objeto está descargando un flowitem.
- **Utilize:** El objeto está siendo utilizado por una estación. Este estado es usado por un operador, cuando llega a una estación y se le requiere para un proceso, set up o alguna reparación.
- **Blocked:** El objeto que está viajando por alguna red es bloqueado por la misma red y no puede avanzar.

❖ Elevator o Elevador:

- Tiene los mismos estados que Operator y Transporter.

### 3.8 CONEXIÓN DE PUERTOS

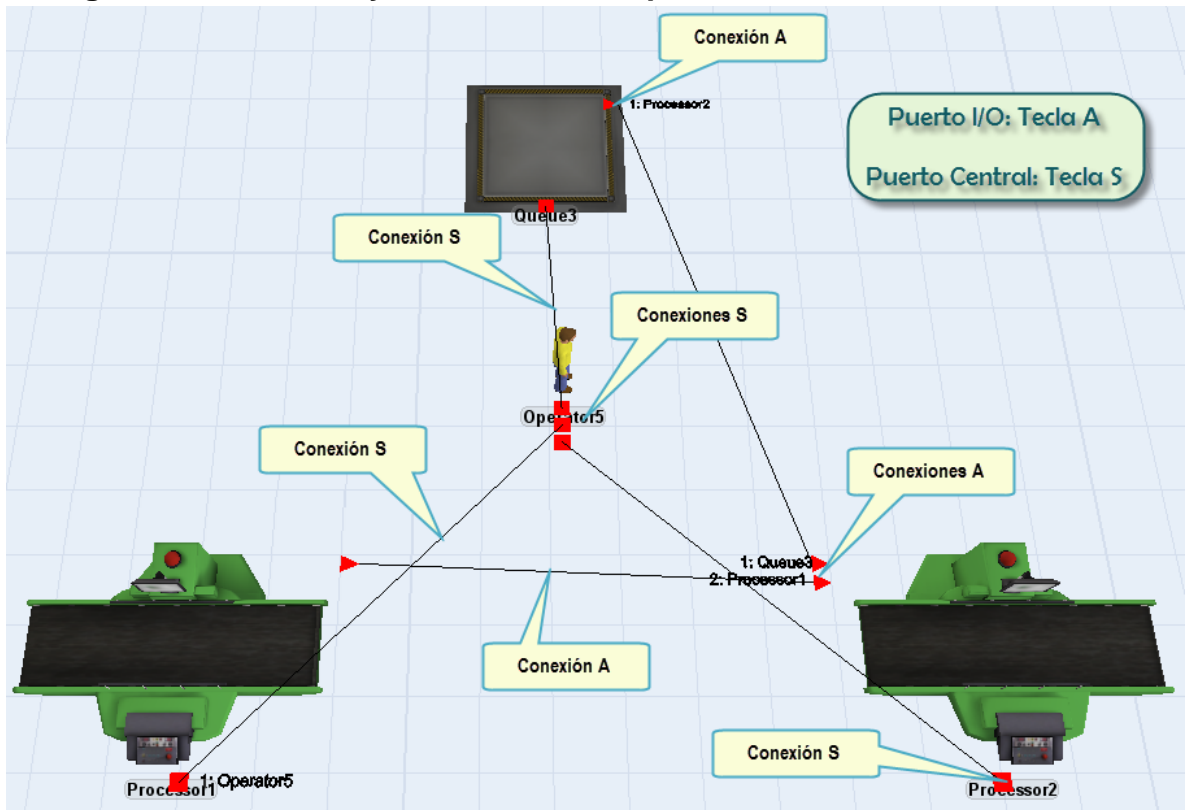
Para que los flowitems se muevan a otro recurso fijo, es necesario definir las conexiones de puertos entre objetos, por lo tanto la conexión se realiza por un puerto de salida del primer objeto hacia un puerto de entrada del segundo objeto, para lo cual se da clic izquierdo en el primer objeto y arrastrando el mouse mientras se oprime la tecla “A” hacia el segundo objeto, de esta manera queda creada la conexión tipo A entre esos dos objetos. En este tipo de conexión siempre es importante el orden en que se realiza, de Salida a Entrada.

Figura 69. Teclas para conexión y desconexión de puertos

	Salida a Entrada	Central a Central
Desconectar	Q U*	W I*
Conectar	A J*	S K*

**\* Son teclas opcionales para zurdos**

Figura 70. Conexión y desconexión de puertos



Las conexiones centrales se hacen con la tecla “S”, manteniéndola oprimida mientras se arrastra el mouse del primer objeto al segundo. Como esta conexión involucra dos puertos centrales, el orden no importa. Se utiliza para asignar recursos móviles u objetos pertenecientes a la clase *Task Executor* (operator, transporter, etc.) a recursos fijos para realizar operaciones y/o transportes.

Los objetos pueden tener un número ilimitados de conexiones de puerto. Los indicadores rojo y verde en los puertos muestra su condición actual como cerrado o abierto. Por ejemplo, un procesador tendrá un indicador de puerto de entrada rojo si está ocupado o no puede recibir un flowitem.

### 3.9 MOVIMIENTO DE FLOWITEMS

Una vez creadas las conexiones para indicar como circulan los flowitems a través de la simulación, debe indicarse la forma exacta en que se mueven. En Flexsim, los flowitems pueden moverse de un objeto fijo a otro recurso a través de las siguientes formas:

- ❖ Inmediatamente: movimiento instantáneo de los flowitems entre los recursos fijos conectados. La conexión es directa entre los objetos y se utiliza la tecla A.
- ❖ Mediante un conveyor: es una banda o cinta transportadora que se utiliza para mover flowitems a través del modelo a lo largo de un camino. Debe deshacerse la conexión entre los recursos fijos con la tecla Q y realizarse la conexión entre el primer objeto (puerto de salida) y el Conveyor (puerto de entrada) con A y después entre el Conveyor (puerto de salida) y el segundo recurso fijo (puerto de entrada) nuevamente con la tecla A, dado que solo se están conectando FixedResources.

Figura 71. Pestaña Conveyor

Operation	Visual
<input checked="" type="checkbox"/> Accumulating	Texture <input type="text" value="fs3d\Conveyor\RollerConveyor: ..."/>
Speed <input type="text" value="1.00"/>	Texture Length <input type="text" value="5.00"/>
Maximum Content <input type="text" value="1000.00"/>	Product Z Offset <input type="text" value="0.00"/>
Spacing Value <input type="text" value="1.00"/>	<input type="checkbox"/> Move While Paused
Spacing Rule <input type="text" value="Item Size"/>	<input type="checkbox"/> Infinite Increment
Spacing Orientation <input type="text" value="Item X Size"/>	Texture Increment: 1 / <input type="text" value="32.00"/>
Orient Z <input type="text" value="0.00"/>	<input checked="" type="checkbox"/> Side skirt follows contour of conveyor (not floor)
Orient Y <input type="text" value="0.00"/>	Side skirt dimension <input type="text" value="0.30"/>
Virtual Length <input type="text" value="0.00"/>	Side skirt offset <input type="text" value="0.10"/>
<input type="checkbox"/> Scale Product Size with Virtual Length	<input type="checkbox"/> Leg base relative to conveyor (not floor)
<input type="checkbox"/> Notify Upstream of Blocked Length	Leg base dimension <input type="text" value="0.00"/>

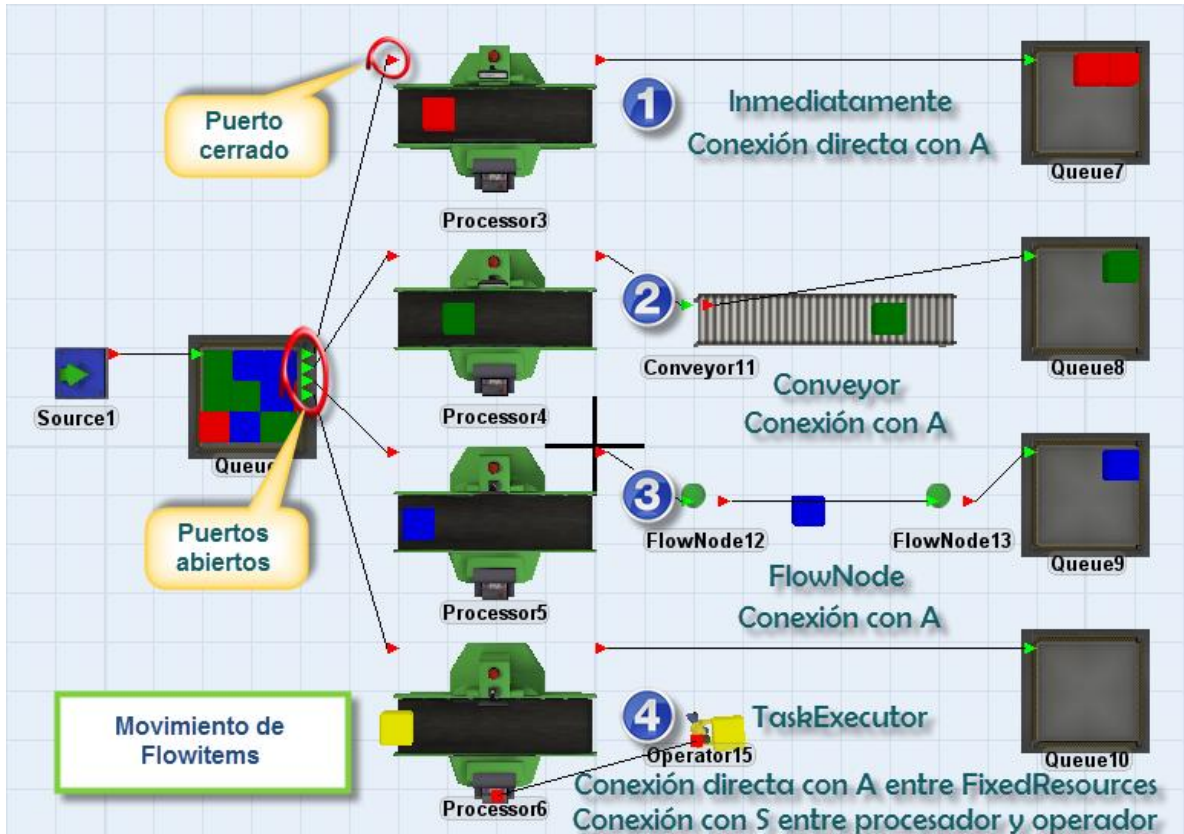
Fuente: Interfaz de Flexsim.

En la pestaña de configuración del conveyor, se pueden ajustar parámetros tales como: su velocidad, longitud, si es de acumulación o no, contenido máximo, etc.

- ❖ Por medio de un FlowNode o nodo de flujo: son recursos fijos que pueden actuar de una manera similar a una muy simple cinta transportadora de acumulación, pero sin piezas visibles. Debe deshacerse la conexión entre los recursos fijos con la tecla Q y realizarse la conexión entre el primer objeto (puerto de salida) y el FlowNode (puerto de entrada) con A, luego entre los FlowNodes con A (si hay uno o varios) y después entre el último FlowNode (puerto de salida) y el segundo recurso fijo (puerto de entrada) nuevamente con la tecla A, dado que solo se están conectando FixedResources.

- ❖ A través de un Task Executer: son recursos móviles que transportan flowitems de un lugar a otro. La conexión entre los recursos fijos es directa y con la tecla A. La conexión entre un recurso fijo y el TaskExecuter es directa y con tecla S a través de los puertos centrales.

Figura 72. Movimiento de Flowitems



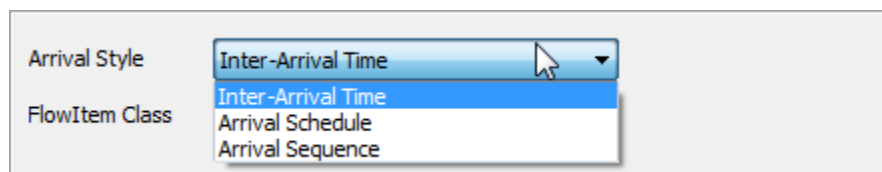
## 4. ADICIONANDO LA LÓGICA DEL MODELO

En el capítulo anterior se explicaron los conceptos básicos de Flexsim, los elementos de un modelo de simulación y como construir objetos y conectarlos. Este capítulo analizará algunas de las propiedades de los objetos en más detalle y explicará cómo agregar la lógica del modelo para que se asemeje al comportamiento de un sistema.

### 4.1 CONFIGURACIÓN DE LLEGADAS Y TIEMPOS DE OPERACIÓN

Lo primero que se define o configura en los modelos de simulación son las llegadas y los tiempos de operación. Las llegadas se definen en el objeto Source, que es donde se crean los flowitems, una vez en sus propiedades, en la sección Arrival Style se tienen 3 opciones:

**Figura 73. Tipos de llegada**

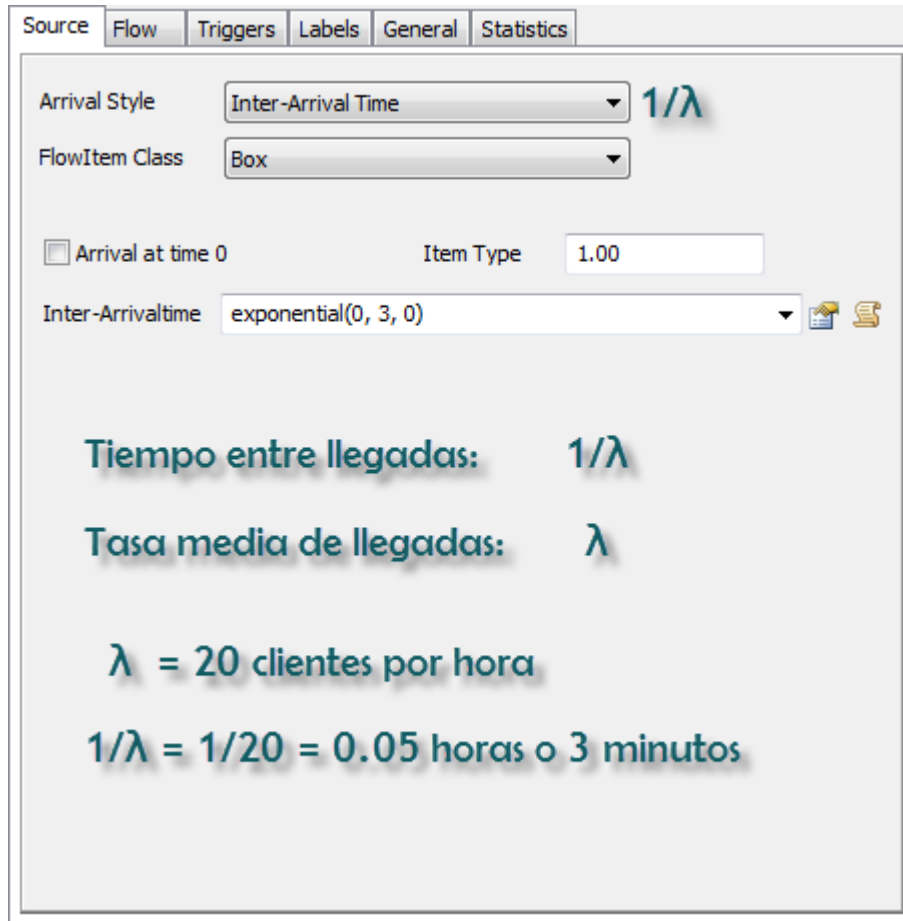


Fuente: Interfaz de Flexsim.

La primera opción es el tiempo entre llegadas, es decir, el tiempo transcurrido entre una llegada y la siguiente. La segunda opción hace referencia a llegadas programadas donde se define en una tabla los tiempos de arribos y sus cantidades y la tercera opción es para definir una secuencia de arribos.

A continuación se explica el tiempo entre llegadas:

Figura 74. Tiempo entre llegadas



Source Flow Triggers Labels General Statistics

Arrival Style Inter-Arrival Time  $1/\lambda$

FlowItem Class Box

Arrival at time 0 Item Type 1.00

Inter-Arrivaltime exponential(0, 3, 0)

Tiempo entre llegadas:  $1/\lambda$

Tasa media de llegadas:  $\lambda$

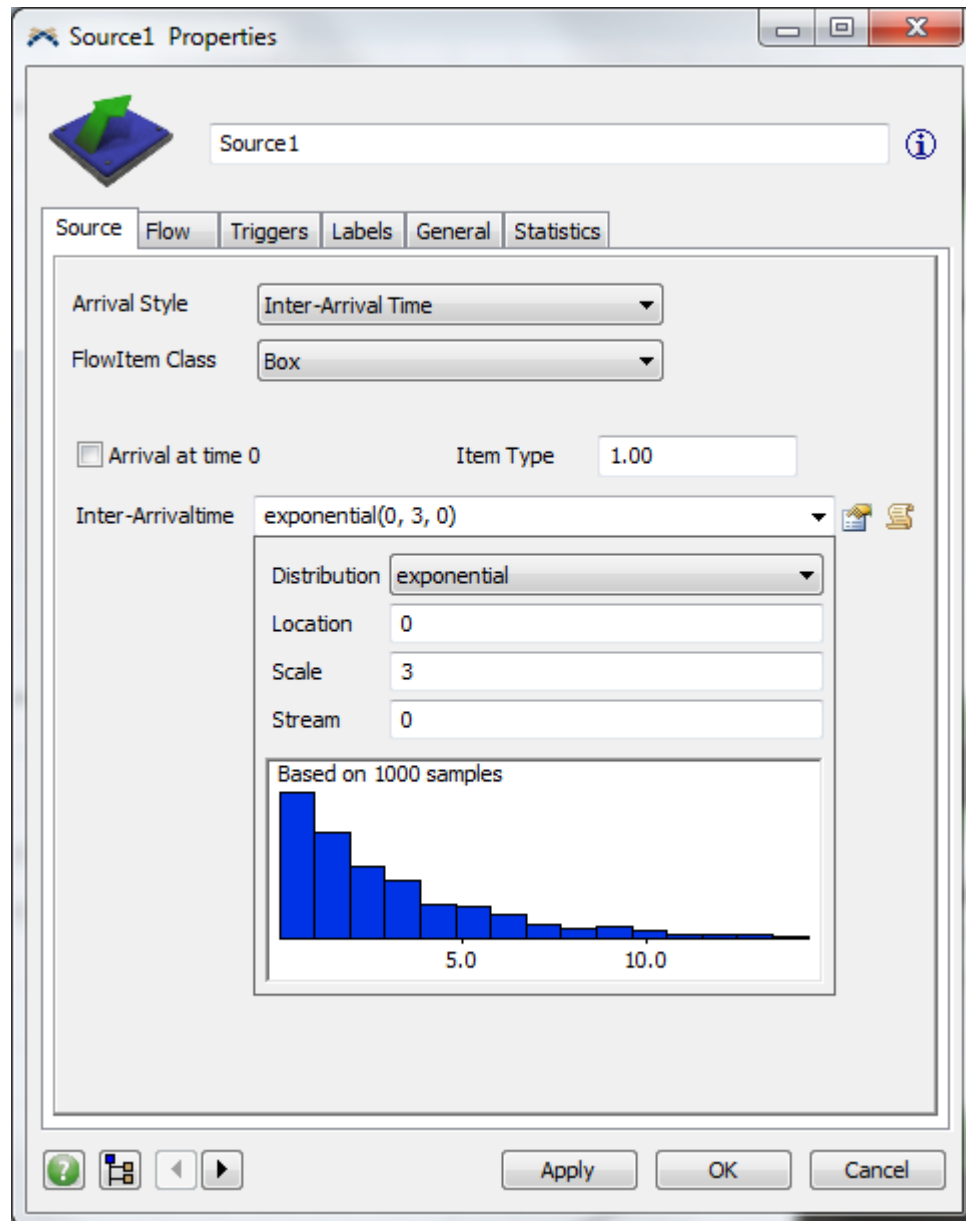
$\lambda = 20$  clientes por hora

$1/\lambda = 1/20 = 0.05$  horas o 3 minutos

Hay que tener en cuenta si se desea que la primera entidad o flowitem llegue al sistema en el primer instante de la simulación o minuto 0.

No se debe confundir la tasa media de llegadas con el tiempo entre llegadas.

Figura 75. Asignación del tiempo entre arribos



Fuente: Interfaz de Flexsim.

Posteriormente se definen los tiempos de operación de cada uno de los centros de trabajo.

Figura 76. Tiempo de operación

Processor Breakdowns Flow Triggers Labels General Statistics

Maximum Content: 1.00  Convey Items Across Processor Length

Setup Time: 0  Use Operator(s) for Setup Number of Operators: 1.00  
 Use Setup Operator(s) for both Setup and Process

Process Time: exponential(0.0, 2.4, 0)  $1/\mu$   Use Operator(s) for Process Number of Operators: 1.00

Pick Operator: centerobject(current, 1) Priority: 0.00 Preemption: no preempt

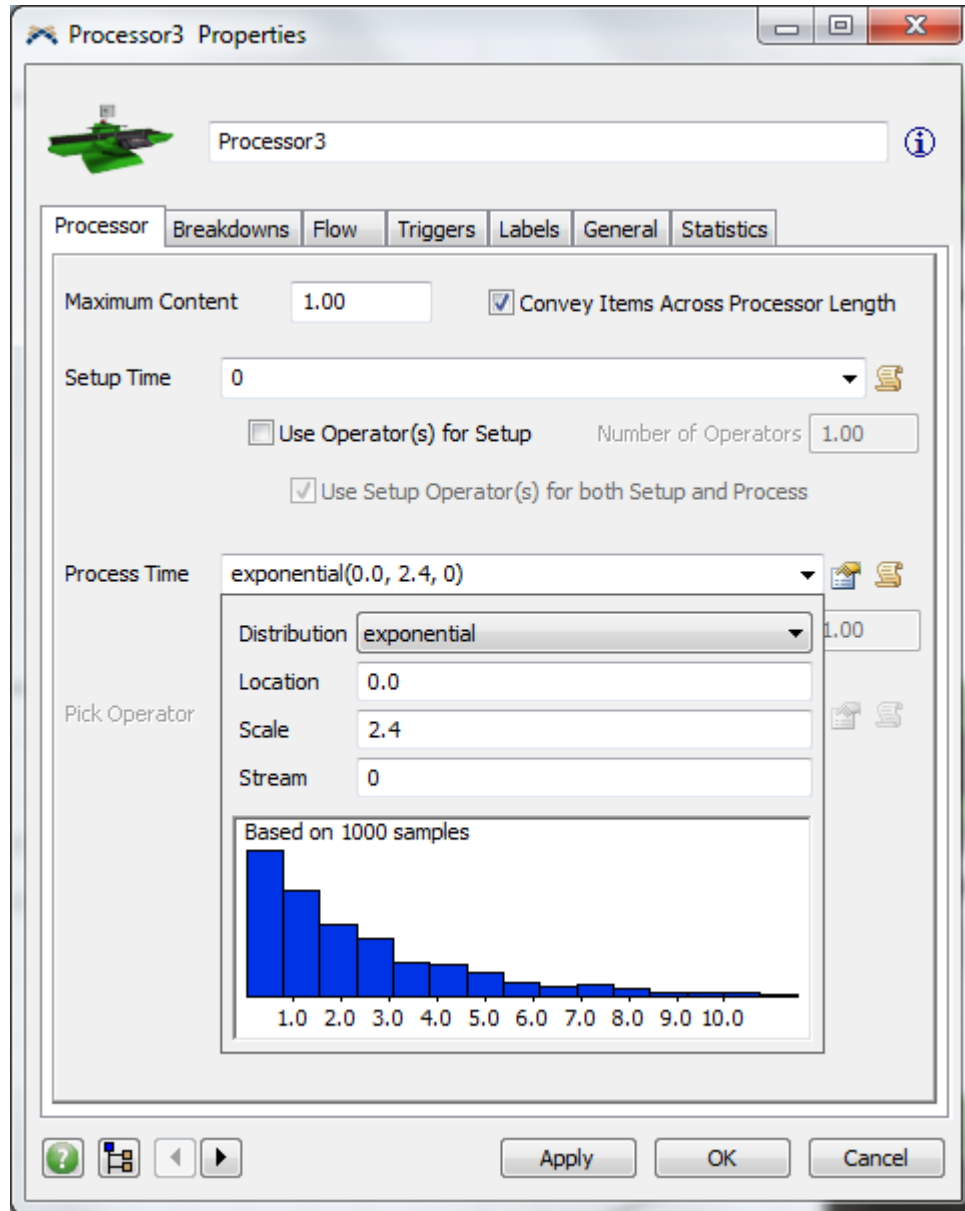
**Tiempo esperado de servicio:  $1/\mu$**

**Tasa media de servicio:  $\mu$**

**$\mu = 25$  clientes por hora**

**$1/\mu = 1/25 = 0.04$  horas o 2.4 minutos**

Figura 77. Asignación del tiempo de operación



Fuente: Interfaz de Flexsim.

## 4.2 ASIGNANDO ATRIBUTOS A OBJETOS

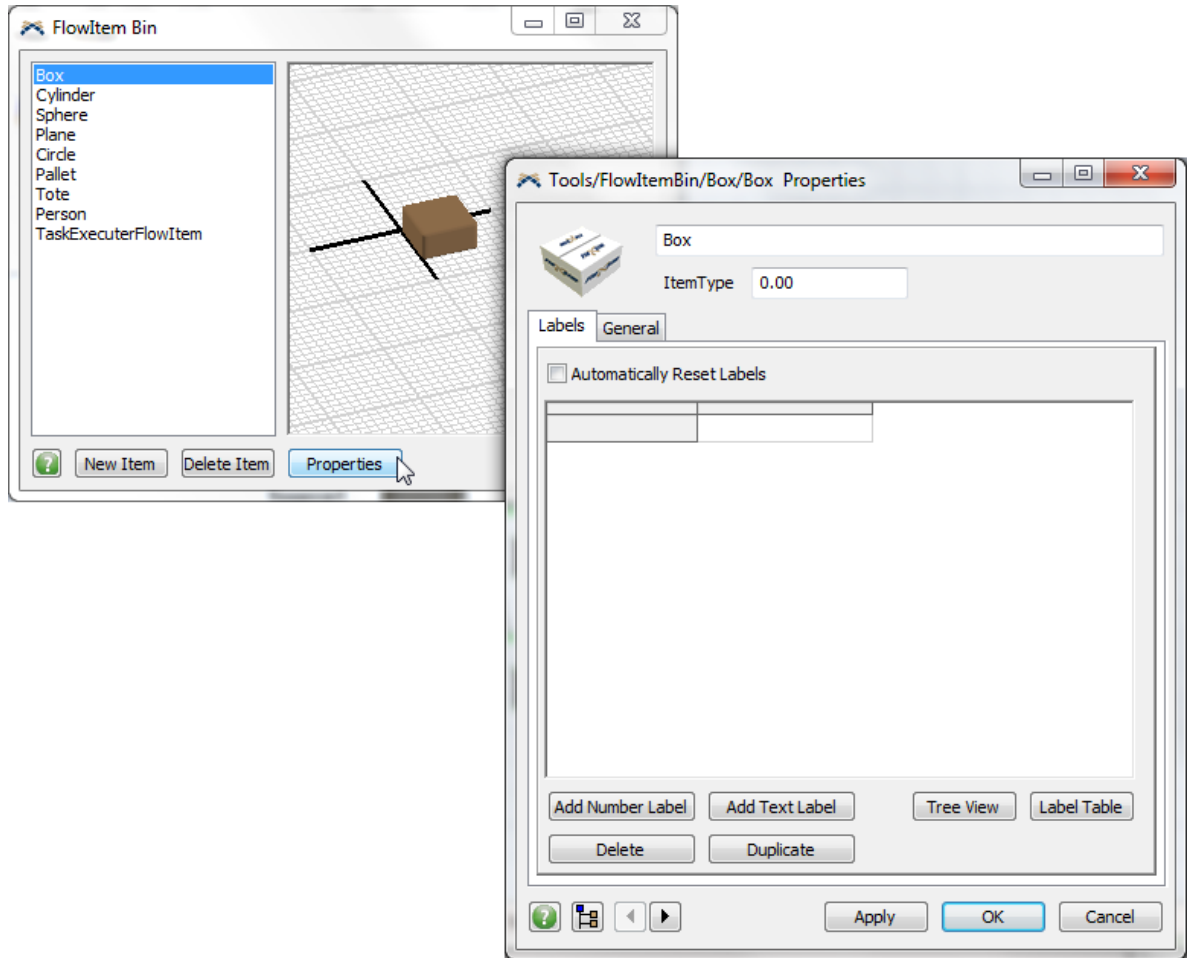
Los atributos son características intrínsecas de un objeto, en Flexsim son llamados etiquetas o labels y permiten mantener información relevante acerca de un objeto. Los atributos pueden ser de tipo numérico o de texto, también admite datos como tablas, siendo una funcionalidad más avanzada para gestionar dicha información.

Las etiquetas pueden ser consultadas desde cualquier parte del modelo y en cualquier instancia durante su ejecución. Se pueden cambiar sus valores directamente desde la pestaña *Labels* de las propiedades o cambiar dinámicamente. Pueden servir como indicadores o banderas de un evento, del estado actual o posición de un objeto o flowitem.

Todos los flowitems tienen una etiqueta numérica por defecto llamada *Itemtype* o tipo de ítem que puede ser cambiada dinámicamente. Por lo general se utilizan para diferenciar el tipo de producto, pieza, materia prima que arriba al sistema o se encuentra en proceso, para definir el flujo de las entidades, diferenciar el tipo de cliente, etc.

Si se desea crear una etiqueta global para un flowitem, se utiliza el contenedor de flowitems o *FlowItem Bin* en el menú Tools y se ingresa a las propiedades de la entidad a editar (véase *Figura 41*), posteriormente se crean los atributos necesarios en la pestaña *Labels*; para los demás objetos solamente se ingresa a las propiedades de estos, dando doble clic y seleccionando la pestaña *Labels* (véase *Figura 64*).

**Figura 78. Pestaña Labels de un Flowitem**



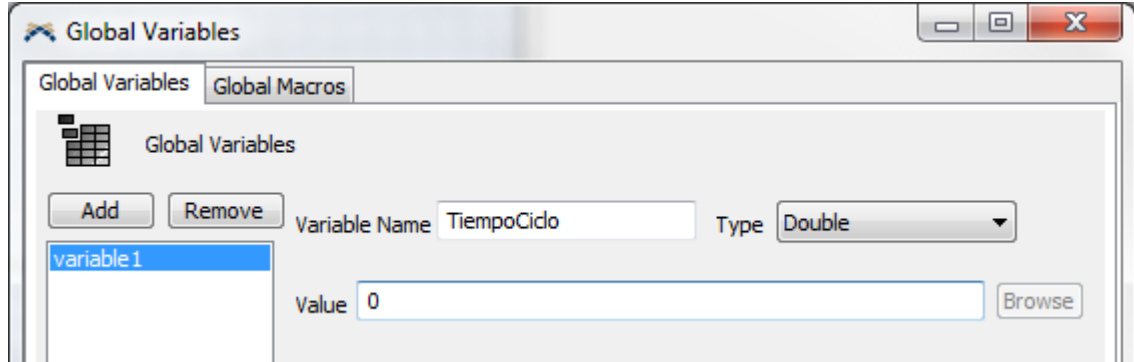
La diferencia radica en que los flowitems se crean en el objeto Source por lo tanto antes de ejecutar la simulación no existen flowitems, así que no se puede ingresar directamente a la pestaña Labels como los demás objetos de la librería.

### 4.3 CREANDO VARIABLES GLOBALES

Las variables son condiciones cuyos valores se crean y modifican. Global se refiere al ámbito o alcance de la variable dentro de Flexsim, al ser global está definida en todo el modelo, si es local, su alcance solamente es la sección donde se definió.

Para su creación se ingresa al menú Tools, opción Global Variables.

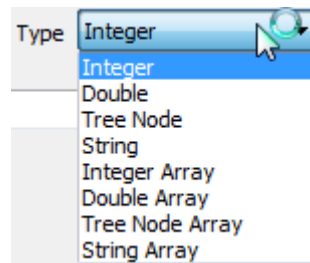
**Figura 79. Creación de variables globales**



En la figura anterior se definió una variable global llamada TiempoCiclo. Es importante mencionar que se debe asegurar de que se inicializan correctamente para que no tomen valores al azar.

Los tipos de datos que puede tomar una variable se encuentran a continuación:

**Figura 80. Tipos de datos**



Double es para tipos de datos reales, Tree Node es un tipo de dato nodo de árbol un poco más avanzado, para este tipo de datos es importante conocer bien la estructura de árbol de Flexsim. El resto de tipos son los comunes, tipo cadena de caracteres o texto, vector de enteros, vector de reales, etc.

#### 4.4 EVENTOS DISPARADORES O TRIGGERS

Los modelos de simulación independiente del lenguaje o software utilizado, requieren de una colección de varios tipos de lógica para representar razonablemente el comportamiento de un sistema de interés, por lo cual suele utilizarse código para programar el comportamiento deseado. Flexsim cuenta con bastante código predeterminado en todos los objetos y en la mayoría de utilidades de las que dispone. Código que sirve para simular comportamientos comunes de un sistema y de fácil acceso para un usuario sin amplios conocimientos en lenguajes de programación. Esta es una característica de los software de simulación especializados.

Esta sección introduce el uso de los **Triggers** de un objeto o disparadores, controladores, que causan un comportamiento específico en respuesta a un evento determinado o cuando se cumple una condición.

Por ejemplo se utilizan para cambiar el color de un objeto, asignar el valor de una etiqueta, cerrar y abrir puertos, para dirigir un flowitem a otro objeto basado en unas condiciones.

**Ejemplo 1.** : Crear 3 diferentes tipos de productos.

Este ejemplo no contempla un proceso elaborado, ni flujos, solamente se centra en las instrucciones adecuadas para crear 3 tipos de items diferentes y explicar el trigger utilizado, en este caso **OnCreation**. Los pasos se detallan a continuación:

1. Crear una fuente de entidades, esto se hace con el objeto *Source* que es el encargado de crear los flowitems.
2. Crear una cola o un Queue.

3. Conectarlos los dos objetos fijos (FixedResources).con la tecla A.
4. Entrar a las propiedades del *Source* con doble clic sobre el objeto, seleccionamos *Inter-Arrivaltime 0* (para que la primera entidad se genere a las cero unidades de tiempo) y escogemos *Statistical Distribution* (distribución estadística) de la lista de selección (*PickList*), para definir un tiempo entre arribos estocástico o probabilístico. exponencial con una media de 3 minutos. Si se desea un tiempo determinístico, solo basta con borrar lo que está escrito en el campo de texto y escribir un valor numérico.

**Figura 81. Ejemplo 1, Paso 4**

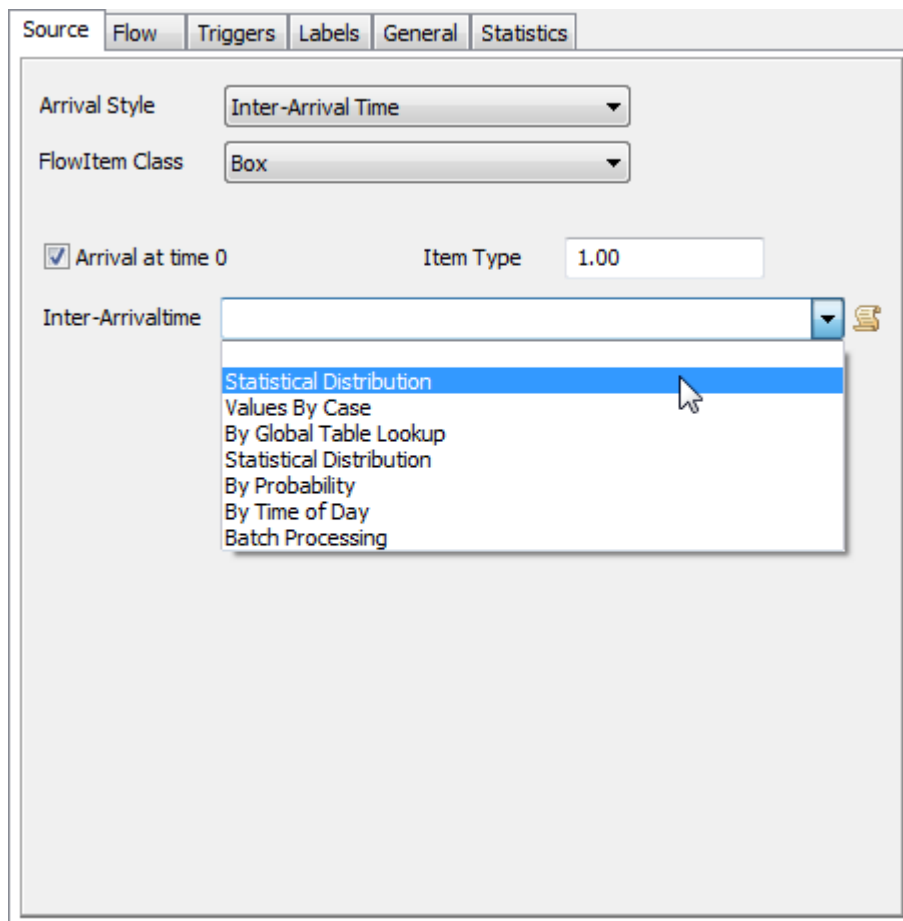


Figura 82. Selección de distribución estadística

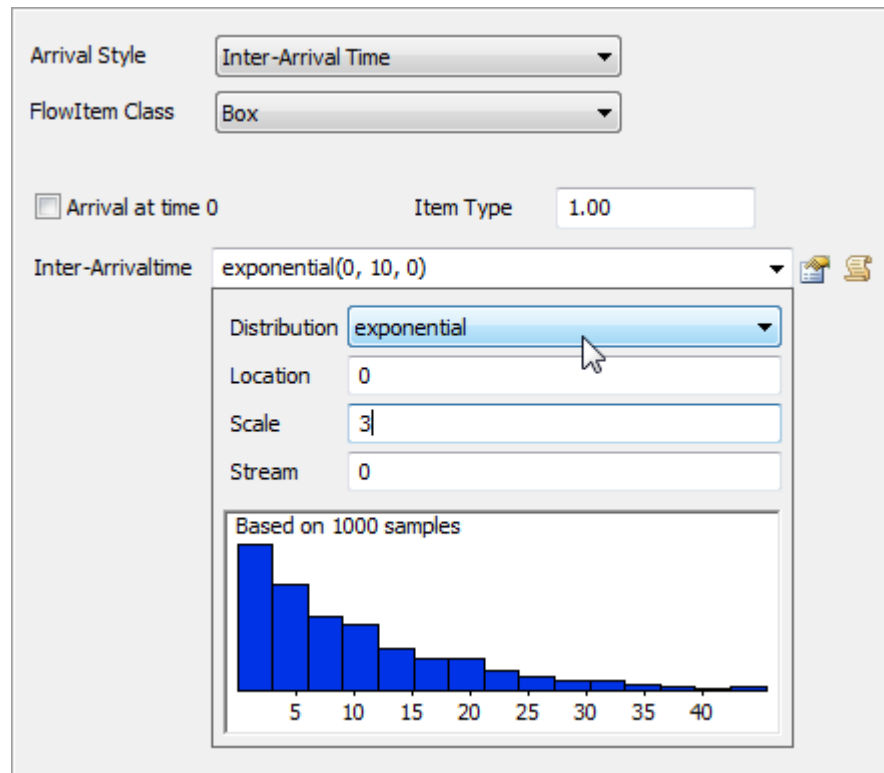
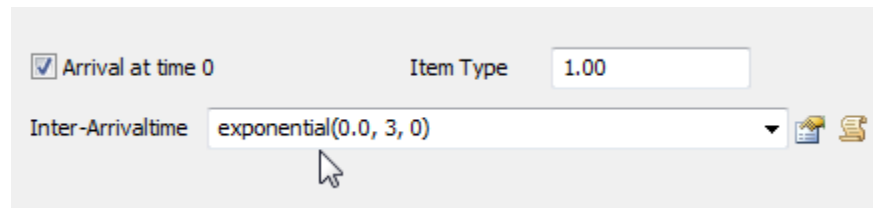


Figura 83. Sintaxis de la distribución exponencial



5. Cambiar a la pestaña *Triggers*, clic en el icono + frente al texto **OnCreation**, aparecerá una lista desplegable y se selecciona el trigger *Set Item Type and Color*, tal como en la figura 77 y 78 a continuación.

Figura 84. Lista desplegable en el trigger OnCreation

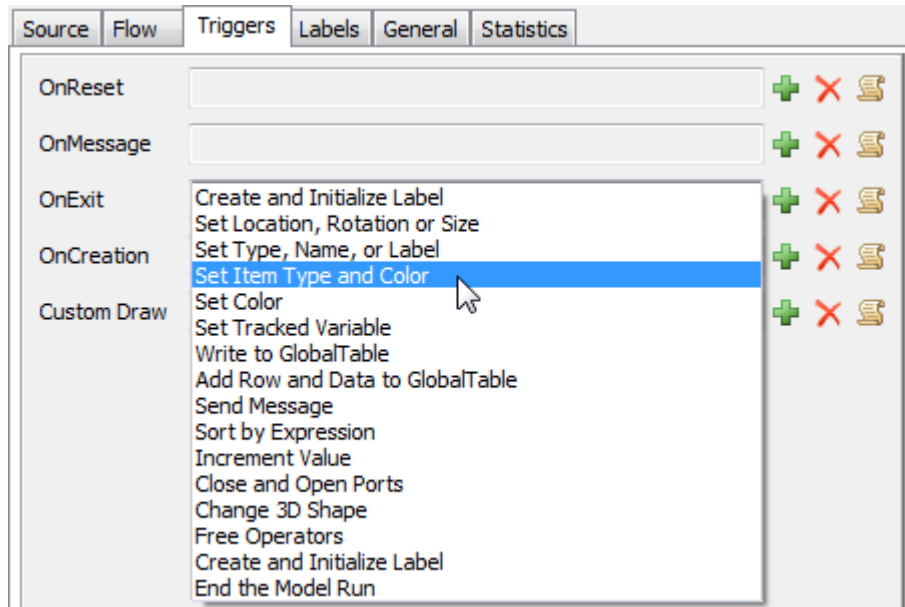
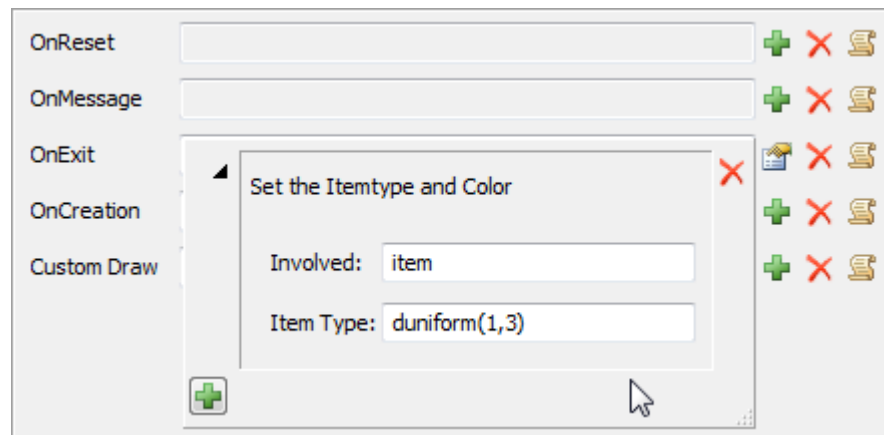


Figura 85. Trigger Set Item Type and Color



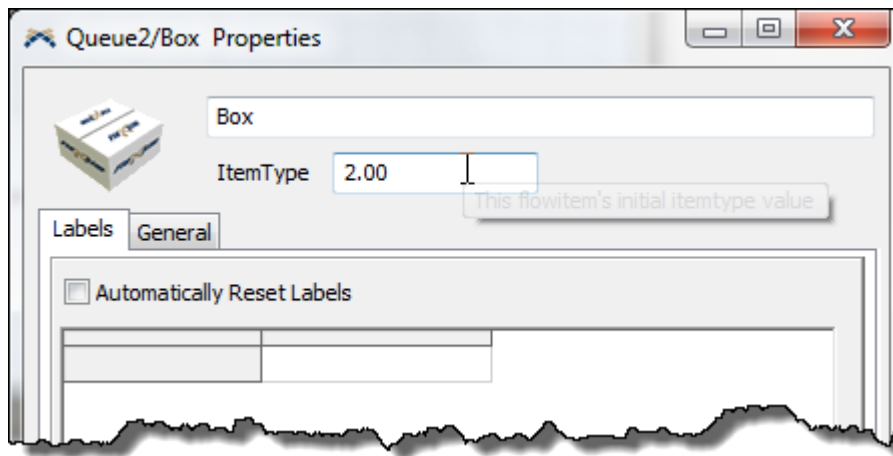
6. Dar clic fuera de ese cuadro contextual y presionar OK.

La distribución uniforme discreta (en Flexsim *duniform*) es similar a la distribución uniforme (*uniform*) excepto que en lugar de arrojar un número real comprendido entre los 2 valores de sus parámetros, solamente arroja números enteros comprendidos en el rango que se le indique, de esta manera a cada

flowitem creado en el objeto *Source* se le asigna un valor entre 1 y 3 distribuido uniformemente para su itemtype.

Presionar *Run* para correr el modelo. Los flowitems que llegan a la cola, llegan de diferentes colores, para ver su Itemtype hay que dar doble clic sobre este.

**Figura 86. Itemtype del flowitem Box**



Los triggers más comunes que se encuentran en la mayoría de objetos son los siguientes:

- ❖ **OnReset:** ocurre cuando el modelo de simulación se reinicia. Es un buen lugar para establecer las condiciones iniciales de las variables y las etiquetas.
- ❖ **OnMessage:** se produce siempre que es enviado un mensaje al objeto. Los mensajes pueden proceder de otros objetos para coordinar la lógica o desde el mismo objeto para crear un tiempo de retardo.
- ❖ **OnEntry:** se produce siempre que un flowitem entra al objeto. A menudo es utilizado para cambiar los valores de las etiquetas de los flowitem como el color, o el Itemtype

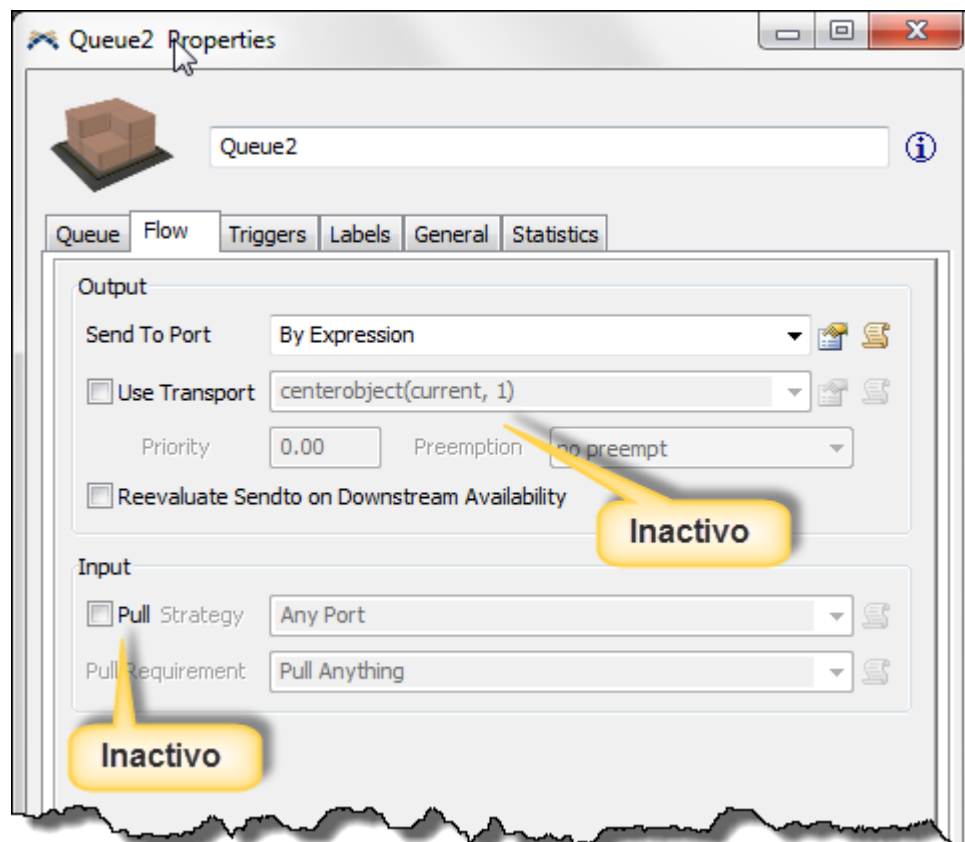
- ❖ **OnExit:** ocurre cuando un flowitem deja al objeto, lo que a menudo se utiliza para actualizar los datos o tomar otra acción antes de que el próximo flowitem entra en el objeto.

Los triggers que se relacionan con el procesamiento, recolección, carga, u otras funciones aparecen en los objetos apropiados.

#### 4.5 FLUJO DEL MODELO

La pestaña de flujo en las propiedades del objeto, hace referencia al movimiento del flowitem actual al siguiente objeto.

**Figura 87. Pestaña de flujo de los objetos**



Como se observa en la figura 80, por defecto esta pestaña tiene deshabilitada las opciones de usar transporte y de entrada (que sólo se activa cuando Pull es seleccionado). La sección de salida (output) por defecto siempre está activa.

❖ **Output:**

- ✓ Send to Port: especifica la lógica que determina a través de que puerto un flowitem puede viajar después de dejar el objeto.
- ✓ Use Transport: cuando está seleccionado llama a un TaskExecuter para transportar el flowitem al siguiente objeto. El menú desplegable se utiliza para especificar transporte debería ser utilizado.

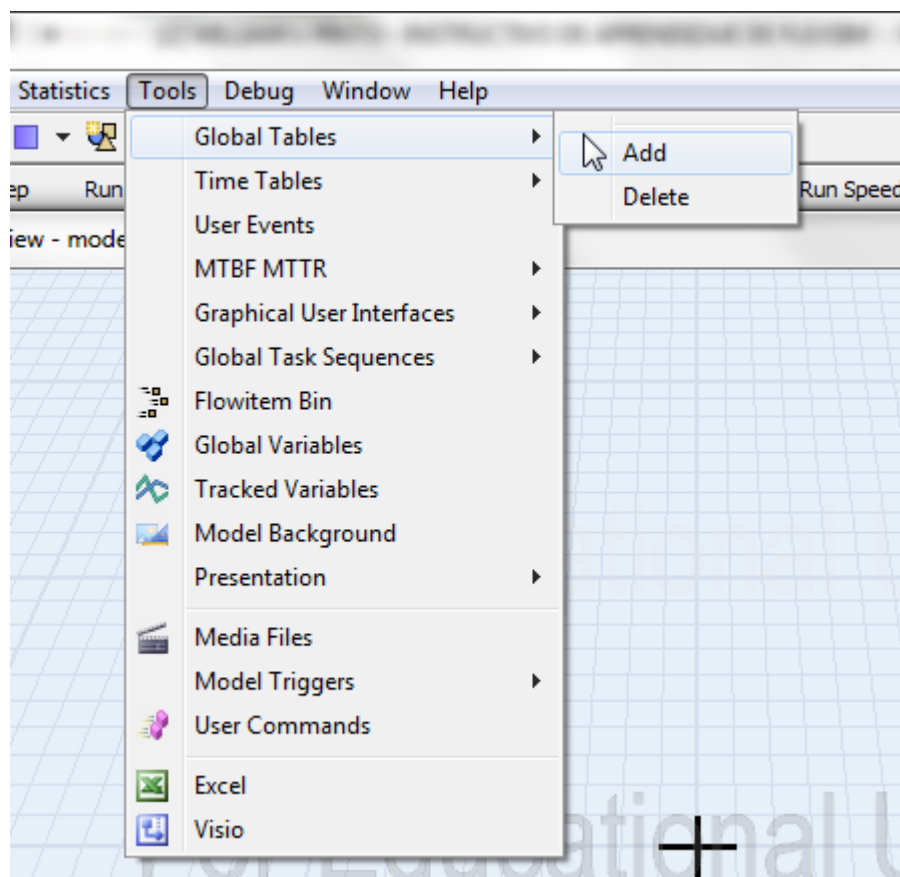
❖ **Input:** la opción Pull es usada cuando los ítems deben ser halados desde el objeto actual en vez de ser empujados (Push) desde el objeto anterior (método por defecto).

- ✓ Pull Strategy: especifica que puerto o puertos deben ser usados para halados.
- ✓ Pull Requirement: especifica la lógica para jalar un flowitem por algún criterio, tal como por el itemtype o por el valor de un atributo.

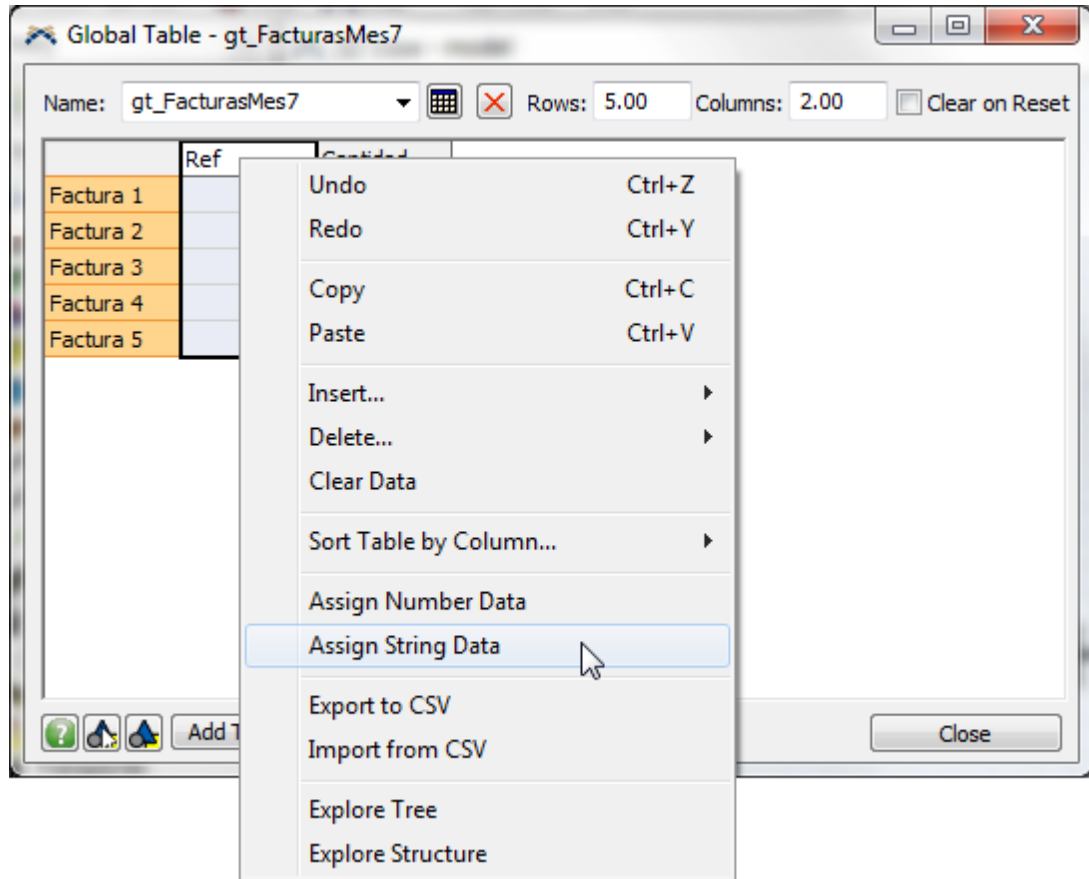
## 4.6 TABLAS GLOBALES

Dado que la administración de los datos del modelo es una de las cuestiones más importantes en los modelos de simulación, una buena práctica es la utilización de tablas globales y su uso depende de la habilidad del modelador. Pueden almacenar cualquier clase de información, como: registros históricos de pedidos, tiempos de procesamiento de productos, tiempos de arribo de clientes, datos de distribuciones empíricas, etc. Su forma es muy similar a las tablas creadas en Excel, aplicación desde la que pueden ser importadas y exportadas. Pueden almacenar datos numéricos o de texto. Las tablas globales son agregadas y editadas a través de la opción *Global Tables* en el menú *Tools*.

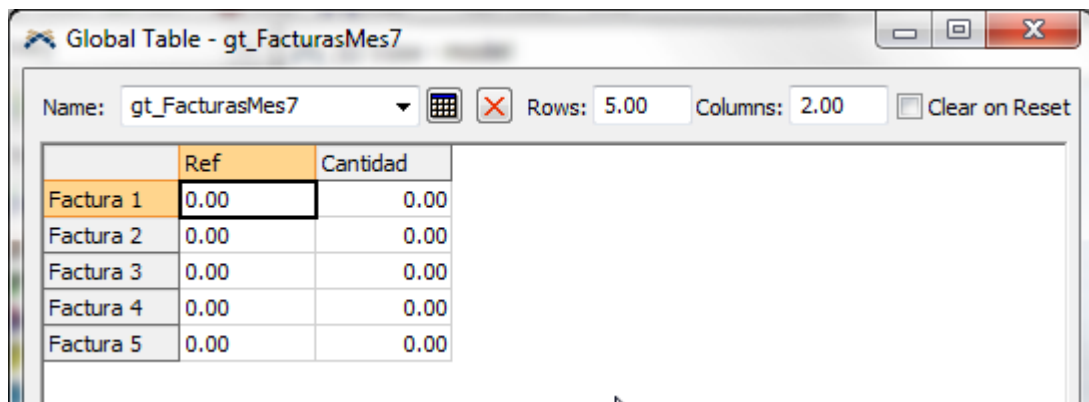
**Figura 88. Tablas Globales**



**Figura 89. Asignando datos de texto en Tablas Globales**



**Figura 90. Ejemplo de Tabla Global**

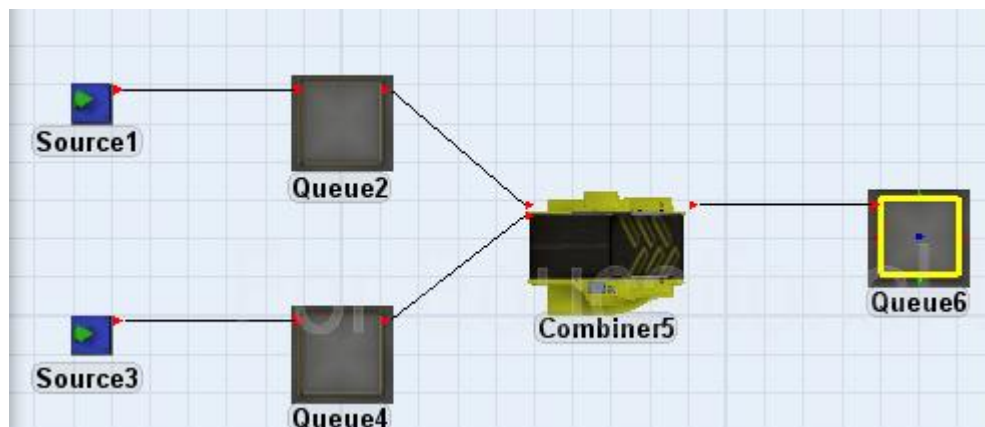


Los siguientes comandos pueden ser utilizados para obtener información de las tablas globales: `gettablenum()`, `gettablestr()`, `settablenum()`, `settablestr()`, `reftable()` que pueden ser consultados en la guía de usuario de Flexsim.

#### 4.7 AGRUPAR Y DESAGRUPAR FLOWITEMS

Para la agrupación y desagrupación de entidades se utilizan los objetos *Combiner* y *Separator*. Un esquema básico de agrupación es el que se muestra a continuación:

**Figura 91. Ejemplo de Combiner**

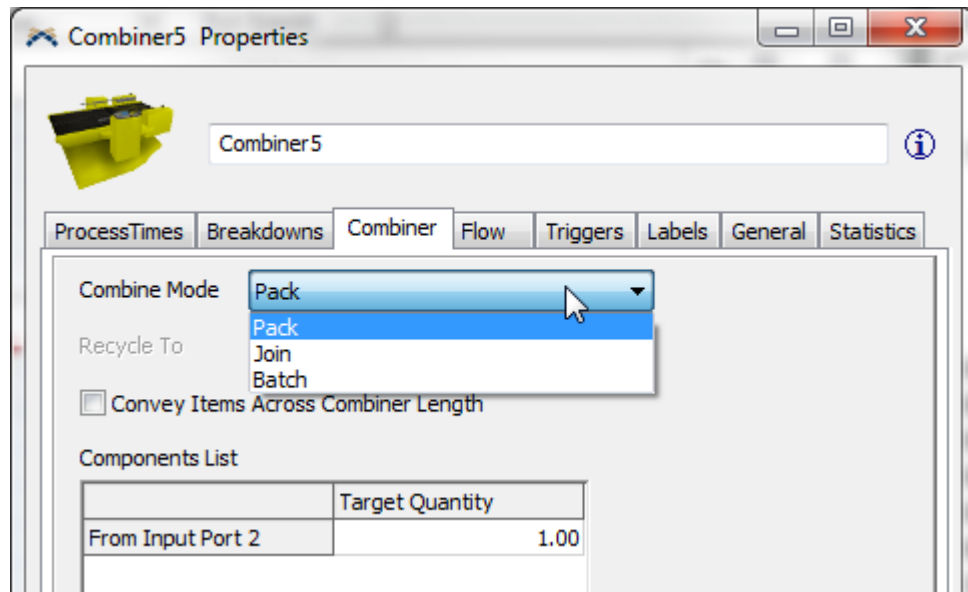


El objeto *Combiner* puede agrupar objetos de una de las siguientes 3 maneras:

- ❖ Pack: modo de empaquetado. El combiner mueve todos los flowitems recibidos por los puertos 2 y superiores dentro del flowitem recibido a través del puerto 1 y luego libera este contenedor de flowitems. Sirve para paletizado.
- ❖ Join: en este modo el combiner destruye todos los flowitems excepto el recibido en el puerto 1. Ensamble permanente de entidades.

- ❖ Batch: modo por lotes. El combiner simplemente libera todos los flowitems una vez que el lote es recogido y los tiempos de setup y de proceso han terminado.

**Figura 92. Modo de combinar flowitems**

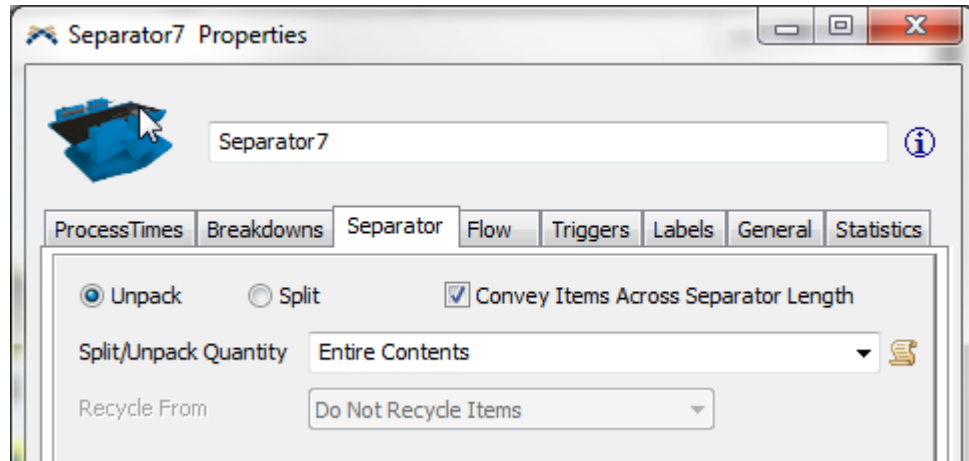


El puerto 1 es el del flowitem base, el 2 y superiores son las entidades a empaquetar en el primer modo de combiner o son las entidades a ensamblar en la entidad base (unión permanente) en el modo Join.

El objeto Separator desempaqueta las entidades que el Combiner empaquetó con la opción Pack. En el caso de Join, no se puede aplicar este objeto para separar entidades.

Otra utilización fuera del desempaqueado, desestibado, es la de procesos de corte o división de flowitems, o cuando se necesita que se dupliquen, para lo cual el Separator debe utilizar la opción Split con el número de entidades a dividir.

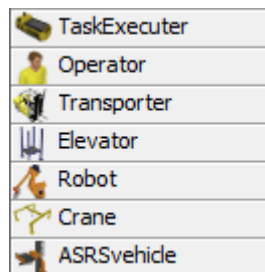
Figura 93. Separator



#### 4.8 UTILIZAR TASK EXECUTERS

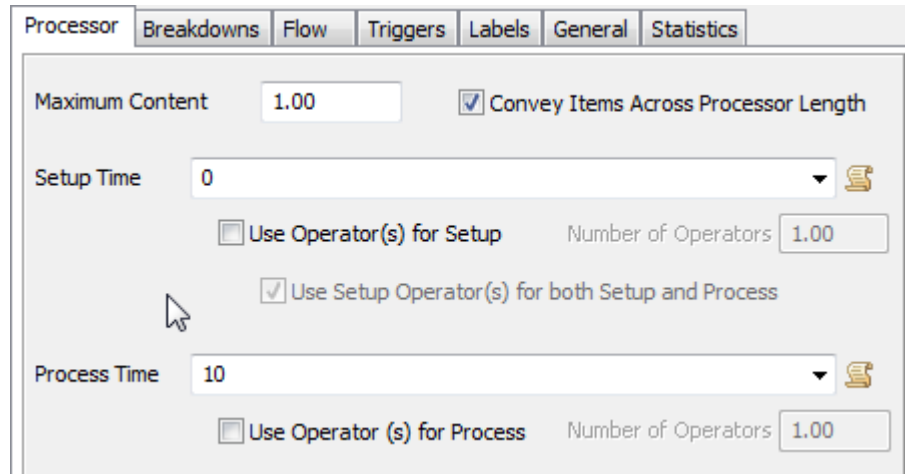
La superclase *TaskExecuter* contiene los siguientes objetos o recursos móviles que son los ejecutores de tareas:

Figura 94. TaskExecuter



Pueden ser asignados para hacer la operación, mantenimiento y alistamiento de los objetos fijos o centros de trabajo.

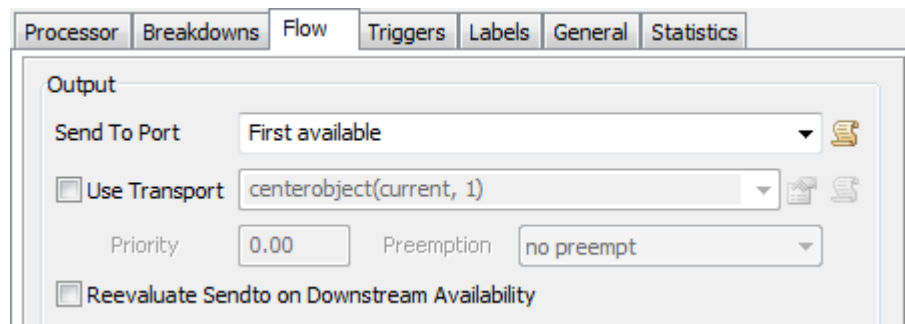
**Figura 95. TaskExecuter**



Tanto en el tiempo de alistamiento o setup, como en el de proceso, si se requiere de un operario que realice estas operaciones, se selecciona la opción: “*use Operator(s) for Setup / Process*”.

Para realizar el transporte de entidades:

**Figura 96. Utilización de operadores**

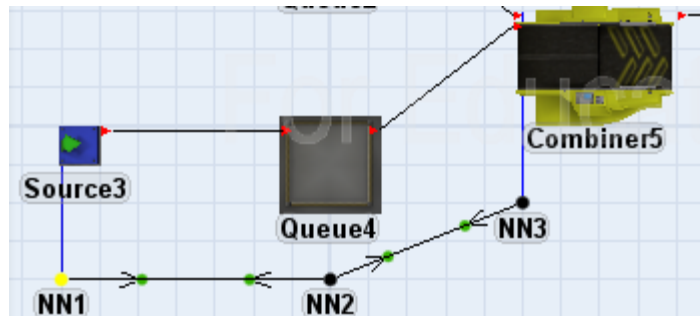


La conexión entre estos objetos y los recursos fijos se hace con la tecla S.

#### 4.9 AGREGAR REDES DE TRANSPORTE

Las redes de transporte son los caminos por los que se transportan las entidades cuando no lo hacen de forma automática o instantánea. Permite que las entidades sigan un recorrido en donde estas no atraviesan otros objetos, paredes, etc.

**Figura 97. Redes de transporte**



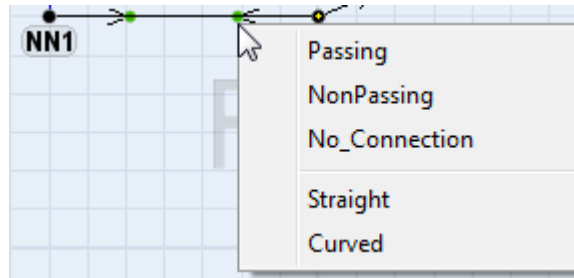
Para su creación se utiliza el objeto Network Node, donde se arrastran los nodos necesarios al layout, se interconectan entre sí con la tecla A, posteriormente se hace la conexión entre el objeto fijo o el TaskExecuter y el nodo con A.

Una línea azul desde el objeto al nodo, indica conexión a objetos fijos, ahora si esta línea es roja, indica que se está conectado a un TaskExecuter.

Los nodos traen su ventana de propiedades, donde se puede configurar parámetros como el tipo de conexión, si es de sobrepaso o no y si es unidireccional o bidireccional, la velocidad permitida y aspectos físicos de los nodos.

Este tema es explicado con mayor amplitud en el taller 3 del presente proyecto de grado. En la imagen a continuación se visualiza el menú contextual del objeto Network Node:

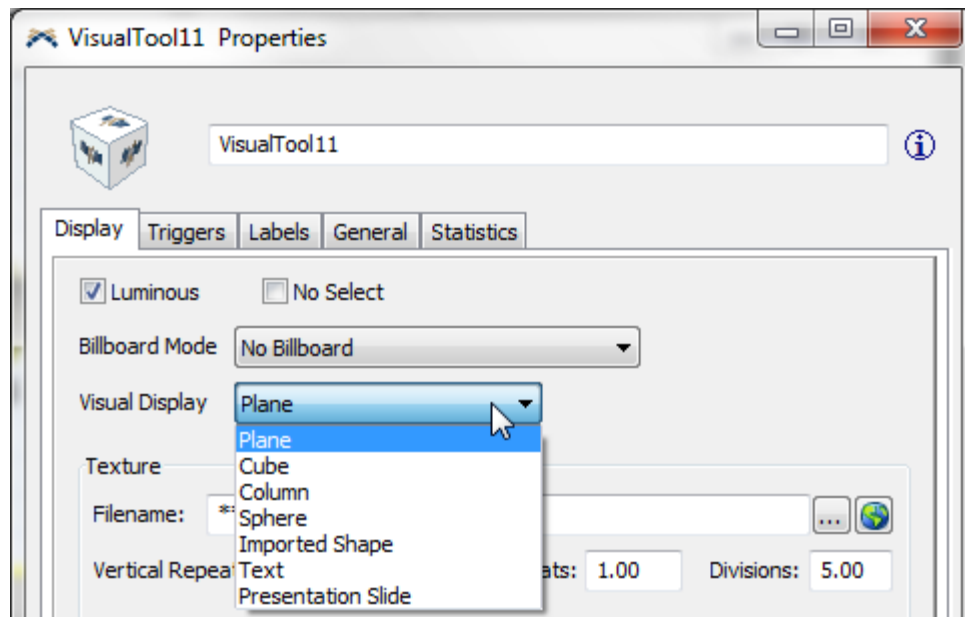
**Figura 98. Tipo de conexión entre nodos**



#### 4.10 VISUALIZAR INFORMACIÓN EN PANTALLA

A menudo es necesario mostrar información en pantalla, tal como el valor de algunas variables globales o el contenido de ciertos atributos, para lo cual se utiliza el objeto *Visual Tool*. Este objeto no realiza acciones directas en la simulación pero provee y reporta información.

**Figura 99. Visual Display**

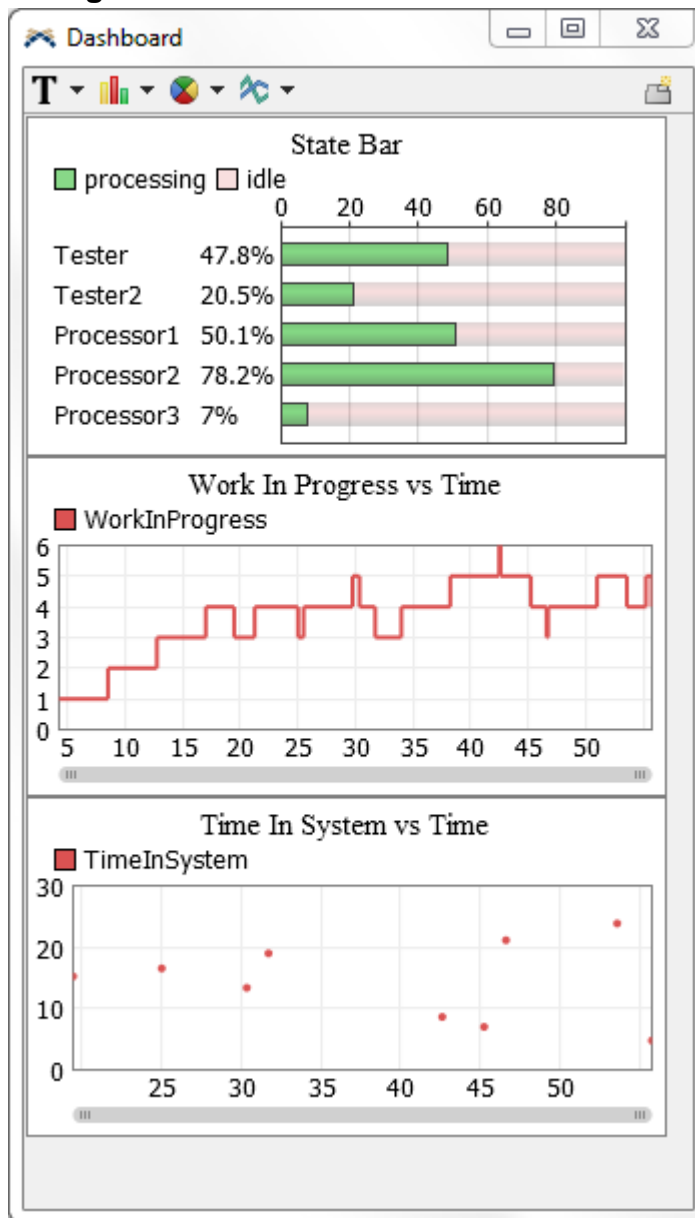


Otro uso de los Visual Tool es el de contenedor de objetos o layout de la simulación.

#### 4.11 DASHBOARD

El *dashboard* es una herramienta que permite agrupar en una ventana pestañas con gráficos e información estadística del modelo para su visualización mientras el modelo está en ejecución. Se puede acceder seleccionando la opción de menú Statistics > Dashboard.

**Figura 100. Dashboard**



Para agregar gráficos en la ventana Dashboard se realizan los siguientes pasos:

1. Seleccionar el tipo de gráfico entre las siguientes 4 opciones: Text Panel, Bar Graph, Pie Chart, Line Graph.
2. Escoger la estadística deseada. La estadística será visualizada para cada objeto en el gráfico.
3. Agregar los objetos de interés.
4. Resetear y correr el modelo.

**NOTA:** No todas las estadísticas tienen sentido para todos los objetos. Si un objeto seleccionado no tiene la estadística especificada, el gráfico no visualizará datos para ese objeto.

La información representada en gráficos o estadísticas puede ser organizada por pestañas en esta herramienta.

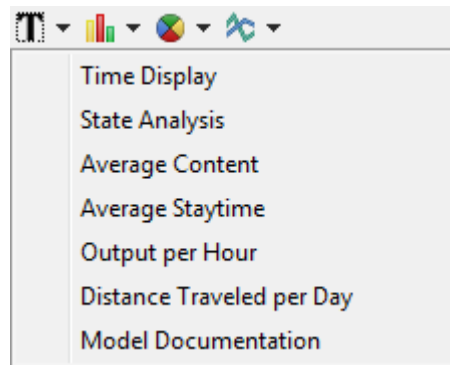
#### 4.11.1 Tipos de Gráficos

##### ❖ Text Panels

- ✓ **Time Display** - Muestra la hora actual del modelo como un momento del día. Le permite seleccionar a qué hora del día inicia el modelo.
- ✓ **State Analysis** – Visualiza el porcentaje de tiempo total que un objeto pasa en un determinado estado. Cada estado se muestra en una columna separada.
- ✓ **Average Content** - Muestra el contenido promedio de los objetos seleccionados.
- ✓ **Average Staytime** - Muestra el tiempo de espera promedio de los objetos seleccionados.

- ✓ **Output per Hour** - Muestra la salida por hora de los objetos seleccionados.
- ✓ **Distance Traveled per Day** - Muestra la distancia recorrida por día de los objetos seleccionados.
- ✓ **Model Documentation** - Muestra un documento HTML que contiene la documentación del modelo.

**Figura 101. Dashboard – Text Panels**



Fuente: Interfaz de Flexsim.

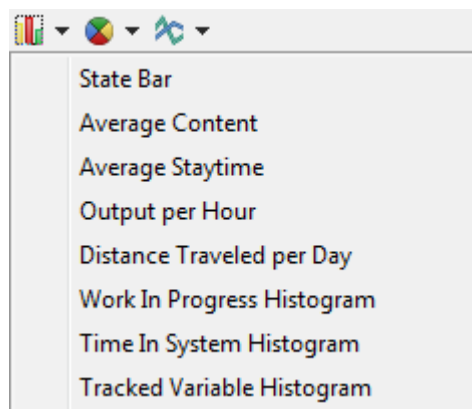
### ❖ Bar Graphs

- ✓ **State Bar** - Muestra el estado de los objetos seleccionados en función del porcentaje.
- ✓ **Average Content** - Visualiza el contenido promedio de los objetos seleccionados.
- ✓ **Average Staytime** - Muestra el tiempo de permanencia promedio de los objetos seleccionados.
- ✓ **Output per Hour** - Muestra la salida por hora de los objetos seleccionados.
- ✓ **Distance Traveled per Day** - Muestra la distancia recorrida por día de los objetos seleccionados.
- ✓ **Work –In Progress Histogram** - Muestra un histograma de la variable de Trabajo en Progreso.

- ✓ **Time In System Histogram** - Muestra un histograma de la variable tiempo en el sistema.
- ✓ **Tracked Variable Histogram** - Muestra un histograma de una variable de seguimiento definida por el usuario.
- ✓ **Tracked Variables**
  - Work In Progress (Trabajo en curso) y Time In System (Tiempo en el sistema) son ejemplos de una variable de seguimiento.
    - **Work In Progress** - Por defecto, es un contador de todos los flowitems en el modelo en un momento dado. Se incrementa cuando se crea un flowitem y se disminuye cuando se destruye un flowitem.
    - **Time In System** - Registra la hora a la que un objeto deja el modelo y el tiempo total que pasó en el modelo.

Las Tracked Variables se pueden crear o personalizar en cualquier momento, utilizando los comandos *settrackedvariable* y *gettrackedvariable*.

**Figura 102. Dashboard – Bar Graphs**



Fuente: Interfaz de Flexsim.

## ❖ Pie Charts

- ✓ **State Pie** - Muestra un gráfico circular o gráfico de pastel de los estados de los objetos seleccionados. Cuando se selecciona más de un objeto, el gráfico muestra el porcentaje medio para cada uno de los estados.

Figura 103. Dashboard – Pie Charts

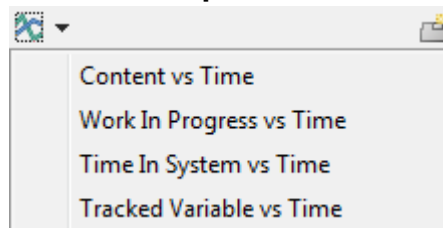


Fuente: Interfaz de Flexsim.

## ❖ Line Graphs

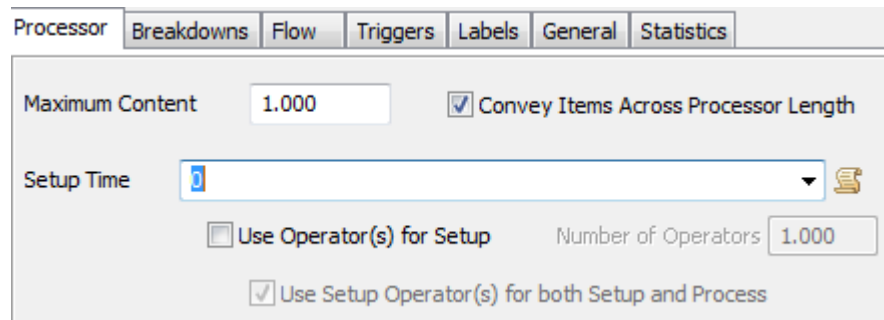
- ✓ **Content vs Time** - Muestra el contenido de los objetos seleccionados contra al tiempo.
- ✓ **Work In Progress vs Time** - Muestra un gráfico de líneas de la variable número total de elementos (ítems) en el modelo contra el tiempo del modelo.
- ✓ **Time In System vs Time** – Grafica el tiempo en el sistema del último flowitem destruido contra el tiempo del modelo.
- ✓ **Tracked Variable vs Time** – Grafica una variable definida por el usuario contra el tiempo del modelo.

Figura 104. Dashboard – Line Graphs




Fuente: Interfaz de Flexsim.

## Picklist

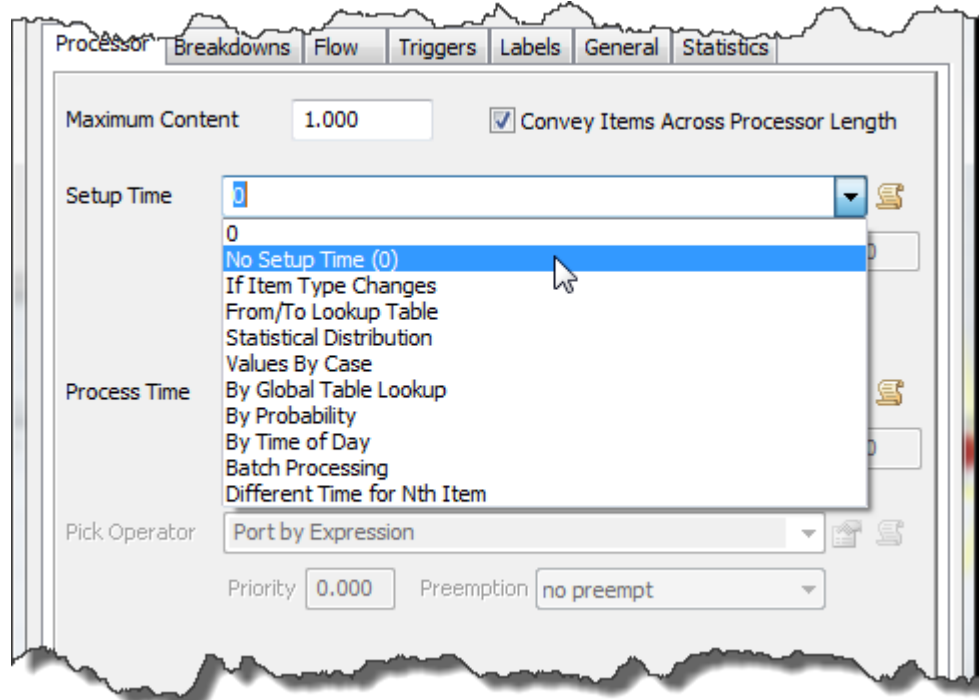


Flexsim tiene bastantes listas desplegables o picklists que permiten ampliar su funcionalidad. a través de piezas de código. Además permiten escribir configuraciones personalizadas sin tener que partir de cero y sin tener que conocer en su plenitud el lenguaje Flexscript.

## Popups

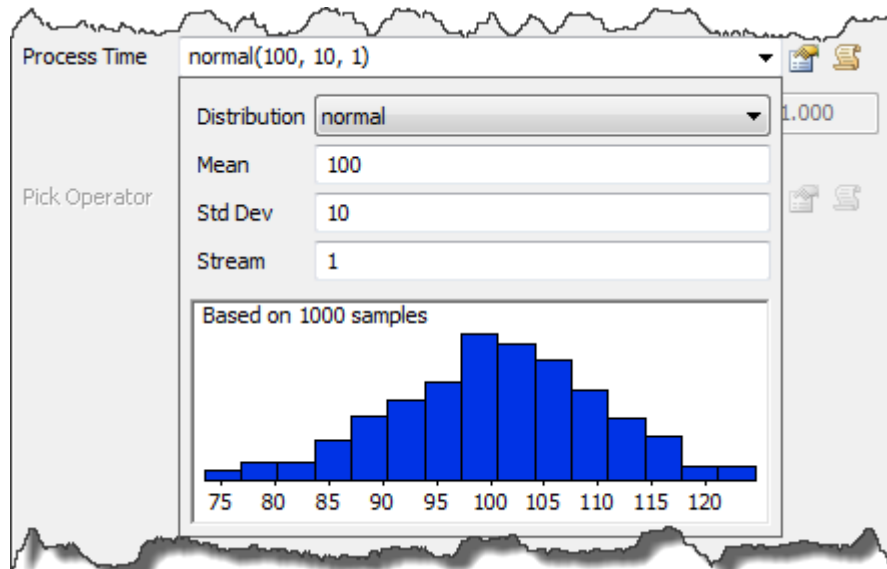
Muchas opciones de la lista desplegable utilizan ventanas emergentes o *popups*. Los *Popups* permiten editar fácilmente los parámetros de la opción escogida en la lista desplegable o *picklist*. Puede modificar estas opciones en cualquier momento haciendo clic en el botón . Una vez que haya introducido los valores que desee en la ventana emergente, haga clic en cualquier lugar fuera de la ventana emergente para cerrarla.

**Figura 105. Listas desplegables o Picklist**




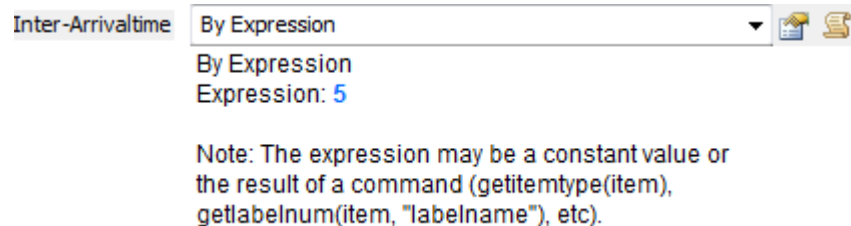
Fuente: Interfaz de Flexsim.

**Figura 106. Ventanas emergentes o popups**



## Code Template

Algunas listas desplegables utilizan plantillas de código. Estas opciones pueden editarse fácilmente cuando se presiona el botón , apareciendo una ventana emergente que explica lo que el código hace y los parámetros que pueden editarse en color Azul.



## Code Edit

Hace referencia a la edición de código Flexscript por parte del modelador. Este es un lenguaje de programación propio de Flexsim, heredado de C++.


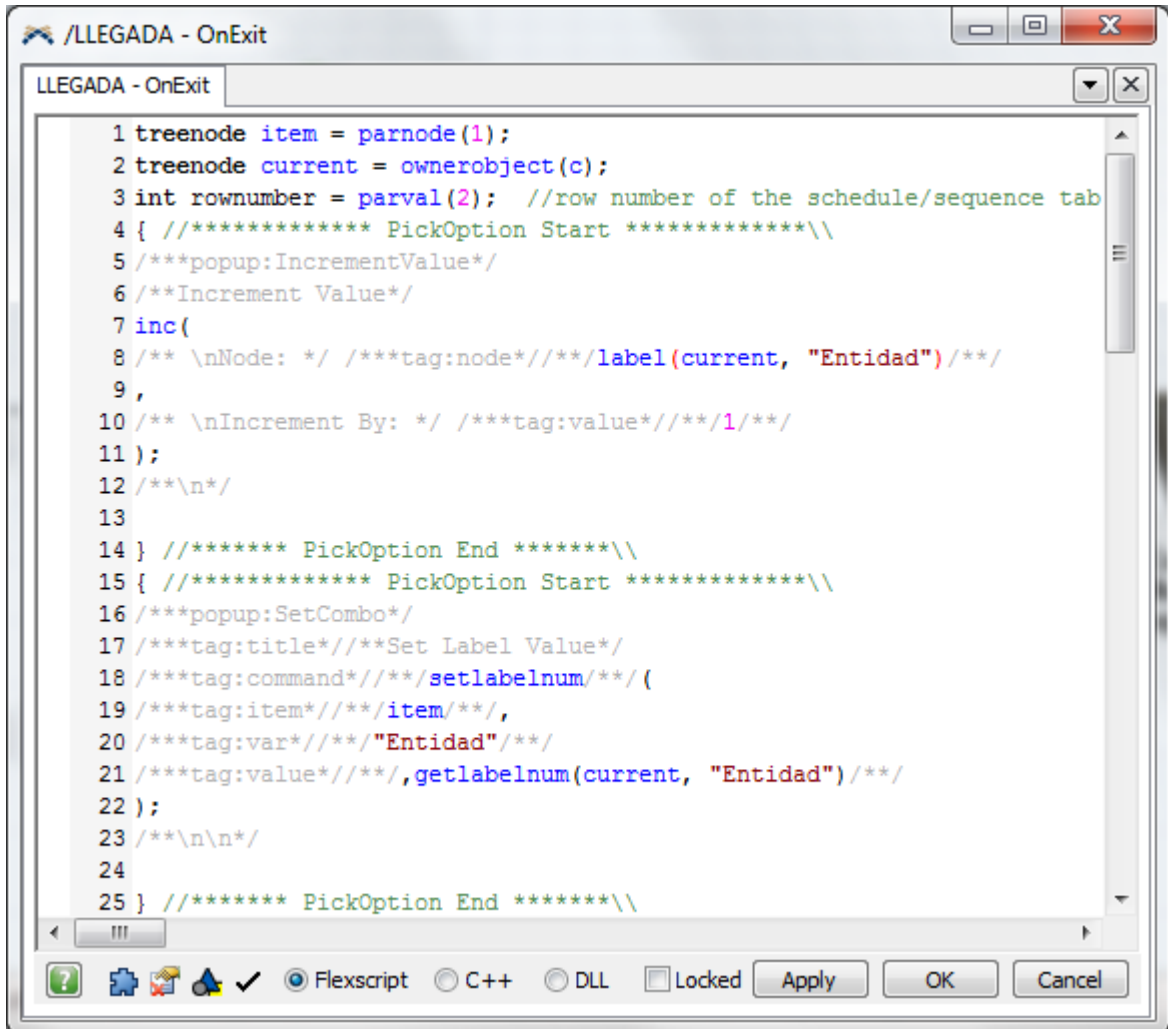
Para acceder a la edición de código se presiona el botón . La mayoría del código es empleado por las plantillas de código para facilitar la labor de usuarios que no estén tan familiarizados con la programación.

Figura 107. Ventana de edición de código



```

1 treenode item = parnode(1);
2 treenode current = ownerobject(c);
3 int rownumber = parval(2); //row number of the schedule/sequence tab
4 { //***** PickOption Start *****\
5 /**popup:IncrementValue*/
6 /**Increment Value*/
7 inc(
8 /** \nNode: */ /**tag:node*//**/label(current, "Entidad")/**/
9 ,
10 /** \nIncrement By: */ /**tag:value*//**/1/**/
11 );
12 /**\n*/
13
14 } //***** PickOption End *****\
15 { //***** PickOption Start *****\
16 /**popup:SetCombo*/
17 /**tag:title*//**Set Label Value*/
18 /**tag:command*//**/setlabelnum/**/(
19 /**tag:item*//**/item/**/,
20 /**tag:var*//**/"Entidad"/**/
21 /**tag:value*//**/,getlabelnum(current, "Entidad")/**/
22 );
23 /**\n\n*/
24
25 } //***** PickOption End *****\

```

## 5. UTILIZACIÓN DE LAS HERRAMIENTAS DE FLEXSIM

Flexsim cuenta con múltiples herramientas o utilidades, entre las cuales se pueden encontrar:

- ❖ **Flexsim Chart:** Una herramienta para la recolección y análisis de datos de salida.
- ❖ **ExpertFit:** Herramienta para el análisis de datos de entrada. Se aborda con detalle en el Taller 3 del Anexo C.
- ❖ **Experimenter:** Módulo para la experimentación del modelo.
- ❖ **OptQuest:** Módulo de optimización.
- ❖ **Herramienta Excel:** Utilidad para la exportación e importación de datos de Excel.

### 5.1 EXPERTFIT

Es un módulo para analizar y determinar el tipo de distribución de probabilidad de un conjunto de datos mediante pruebas de bondad de ajuste. Un objetivo secundario es proporcionar asistencia en el modelado de una fuente de aleatoriedad (por ejemplo, un tiempo de servicio) en ausencia de datos.

ExpertFit se encuentra únicamente en las versiones licenciadas de Flexsim, siendo imposible acceder a este módulo desde una versión *Trial* o de prueba de Flexsim. Para acceder al archivo de ayuda es necesario hacerlo desde esta aplicación, dado que la ayuda de Flexsim no documenta este módulo por ser adicional al software.

El taller 3 del anexo “Talleres prácticos de simulación con Flexsim” explica en detalle esta herramienta para el análisis de datos.

## 5.2 EXPERIMENTER

Es un módulo que abre un cuadro de diálogo que permite ejecutar experimentos corriendo el modelo a través de múltiples escenarios, cambiando ciertas variables entre las réplicas del modelo, y recolectando los datos de salida de cada escenario, donde cada uno representa una configuración diferente del modelo. El modelo se ejecuta para varias repeticiones o réplicas por cada escenario. La herramienta *Experimenter* es accesible desde el menú **Statistics**.

El taller 4 del anexo “Talleres prácticos de simulación con Flexsim” explica en detalle esta herramienta para la simulación de múltiples escenarios.

## BIBLIOGRAFÍA

- GARCÍA, Eduardo; GARCÍA, Heriberto y CARDENAS, Leopoldo. Simulación y análisis de sistemas con ProModel. Pearson Educación, México, 2006.
- BLANCO Rivero, Luis; FAJARDO Piedrahita, Iván. Simulación con ProModel: casos de producción y logística. Escuela Colombiana de Ingeniería, Bogotá, 2003.
- GARAVITO Hernández, Edwin A.; ARENAS Díaz Piedad. Simulación de Procesos de Manufactura 1. Publicaciones UIS, Bucaramanga, 2008
- HARREL, Charles; GHOSH Biman. Simulation using ProModel. First Edition, McGraw Hill, United States of America, 2006.
- KELTON, W. David; SADOWSKI, Randall and SADOWSKI, Deborah. Simulation with Arena. Second Edition, McGraw Hill, United States of America, 2000.
- Página Web de Flexsim Simulation Software:  
<http://www.flexsim.com/>
- Página Web de Flexsim Entrenamiento Online:  
<http://www.competition.flexsim.com/entrenamiento/>
- Sitio Web de ProModel Corporation:  
<http://www.promodel.com/>

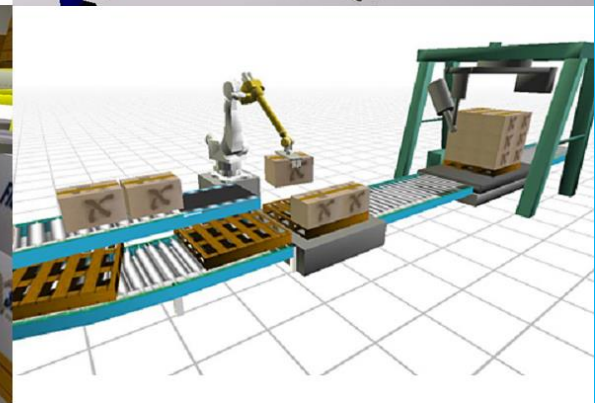
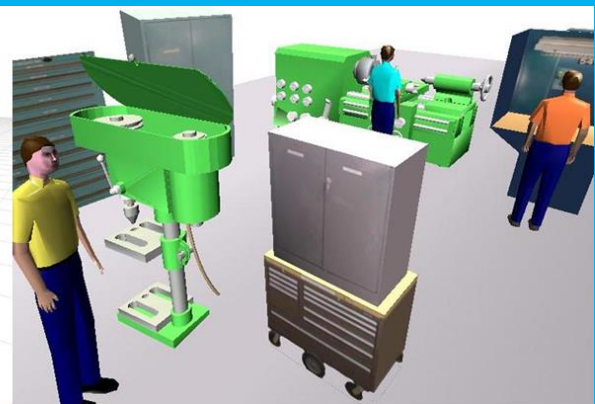
**ANEXO B**  
**Talleres Prácticos con el Software de Simulación Flexsim**



TÉCNICAS MODERNAS DE  
OPTIMIZACIÓN

2013

## TALLERES PRÁCTICOS CON EL SOFTWARE DE SIMULACIÓN FLEXSIM



WILLIAM JESÚS PINTO TRIANA  
Ingeniero Industrial UIS  
EEIE

**TALLERES PRÁCTICOS CON EL  
SOFTWARE DE SIMULACIÓN FLEXSIM**



**WILLIAM JESUS PINTO TRIANA**

**DOCENTE**

**EDWIN ALBERTO GARAVITO HERNÁNDEZ**



**UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS  
ESCUELA DE ESTUDIOS INDUSTRIALES Y EMPRESARIALES  
BUCARAMANGA**

**2013**

## TABLA DE CONTENIDO

	pág.
INTRODUCCIÓN.....	3
MATERIAL COMPLEMENTARIO .....	4
TALLER 1: SIMULACIÓN DE LÍNEAS DE ESPERA .....	5
TALLER 2: FILAS DE ESPERA CON TABLAS GLOBALES .....	35
TALLER 3: ANÁLISIS DE DATOS DE ENTRADA CON EXPERTFIT .....	74
TALLER 4: SIMULACIÓN DE ESCENARIOS.....	101
TALLER 5: SISTEMAS DE CONTROL DE PRODUCCIÓN .....	115
TALLER 6: PROYECTO INTEGRADOR.....	121
BIBLIOGRAFÍA .....	127

## INTRODUCCIÓN

Este manual consiste en 6 talleres correspondientes al empleo del software de simulación Flexsim y sus herramientas. El enfoque dado permite construir modelos paso a paso e ir introduciendo a medida que avanza en su lectura, nuevos elementos de construcción de modelos de simulación.

Los dos primeros talleres tendrán énfasis en modelos de líneas de espera y explicación de las herramientas básicas, el tercer taller utilizará la herramienta *ExpertFit* para el análisis de datos de entrada, el cuarto taller empleará la herramienta *Experimenter* para la simulación de escenarios y análisis de datos de salida, el quinto usará redes de transporte y secuencias de tareas, además comparará dos modelos en paralelo para observar sus diferencias. El último taller integra los conceptos y herramientas vistas en los talleres anteriores y será más de aplicación por ende la complejidad es mayor.

La metodología de trabajo se basa en desarrollar en cada clase un ejercicio guiado por el docente y complementado con un taller que deberá ser elaborado por el estudiante y entregado para su evaluación. Se incluye en el material una seis (6) videos que contienen la construcción y explicación de los ejercicios guiados por el docente para cada taller y archivos en Excel que contienen información necesaria para el desarrollo de los talleres. El material está disponible en la plataforma Moodle de la asignatura.

Salvo que se indique lo contrario, las funciones descritas siempre se referirán a la versión 6.0.2 de Flexsim.

## MATERIAL COMPLEMENTARIO

### VIDEOS

- Taller No. 1 Simulación de Líneas de Espera - Ejercicio.mp4.
- Taller No. 2 Teoría de Colas - Ejercicio.mp4.
- Taller No. 3 ExpertFit - Ejercicio.mp4.
- Taller No. 4 Experimenter - Ejercicio.mp4.
- Taller No. 5 Sistemas de Control de Producción.mp4.
- Taller No. 6 Proyecto Integrador.mp4.

### MODELOS

- Taller No. 1 Teoría de Colas - Ejercicio.fsm
- Taller No. 2 Filas de Espera con Tablas Globales - Ejercicio.fsm
- Taller No. 3 ExpertFit - Taller.fsm
- Taller No. 3 ExpertFit - Ejercicio.efp
- Taller No. 4 Simulación de Escenarios (Experimenter) - Ejercicio.fsm
- Taller No. 5 Sistemas de Control de Producción.fsm
- Taller No. 6 Proyecto Integrador.fsm

### ARCHIVOS EN EXCEL

- Taller No. 2 Filas de Espera con Tablas Globales.xlsx
- Taller No. 3 ExpertFit - Tiempos.xlsx
- Taller No. 6 Tiempos.xlsx

## TALLER 1: SIMULACIÓN DE LÍNEAS DE ESPERA

### INTRODUCCIÓN

Las líneas de espera generan malestar, ineficiencia, retraso y otros problemas, lo que origina un costo de tiempo y económico. Es muy importante evaluar el balance entre el aumento del nivel de servicio y el tamaño de las colas de espera. Por tanto, es necesario entender la relación entre el número de servidores en un sistema (o eficacia de los mismos) y la cantidad de tiempo gastado en la cola (o cantidad de clientes en la misma).

En sistemas de colas sencillos dichas relaciones se pueden encontrar analíticamente y/o simular en una hoja de cálculo. En sistemas más complejos se pueden analizar mediante simulación en un software especializado.

El estudio de teoría de colas es indispensable para comprender conceptos básicos en un simulador como Flexsim, tales como: factor de utilización, tiempo de espera promedio, disciplina de servicio, etc. Este tema permite introducir al estudiante de una manera sencilla en la simulación de eventos discretos en Flexsim.

### OBJETIVOS

- ❖ Construir un modelo de simulación de líneas de espera en Flexsim mediante la configuración de parámetros como tiempos entre llegadas y tiempos de operación determinísticos y estocásticos o probabilísticos.
- ❖ Conocer y aplicar las herramientas del software Flexsim para la solución de eventos relacionados con la teoría de colas.

- ❖ Configurar un sistema básico de colas M/M/1.

## METODOLOGÍA

La práctica consta de dos partes:

- Parte 1: correspondiente al ejercicio guía, que se acompaña de un archivo en Flexsim con el desarrollo del ejemplo y un vídeo explicativo de la construcción del ejercicio.
- Parte 2: correspondiente a los ejercicios que el estudiante debe desarrollar.

Para el ejercicio guiado se simula un sistema de líneas de espera con una cola y con un servidor, donde se aprenderá a introducir un tiempo entre llegadas y un tiempo de servicio determinístico y probabilístico, a configurar la capacidad de una fila y la ubicación de las entidades dentro de esta. También se utilizarán etiquetas para guardar un atributo de interés de un objeto en Flexsim.

## EJERCICIO GUÍA

1. Se tiene un sistema de líneas de espera con una fila y con un servidor. La tasa media de llegadas es de 20 clientes (o productos) por hora y la tasa media de servicio de 25 clientes (o productos) por hora. Simular el sistema por 1000 minutos, con tiempos determinísticos.
2. Emplear un tiempo entre llegadas y de servicio distribuidos exponencialmente con los parámetros del punto 1.

Los pasos necesarios para modelar el sistema se indican a continuación:

Primero se establecen las unidades del modelo y posteriormente se identifican los elementos del sistema. Las unidades están dadas en metros, minutos y litros para longitud, tiempo y fluidos respectivamente.

## OBJETOS DEL MODELO

Los objetos requeridos en Flexsim para el ejercicio guía son los siguientes:

**Tabla 1. Elementos del modelo guía**

Elemento	Objeto	Nombre	Puertos		
			Entrada	Central	Salida
Llegadas	Source	LLEGADA	0	0	1
Filas	Queue	COLA	1	0	1
Operaciones	Processor	SERVIDOR	1	0	1
Salidas	Sink o Queue	SALIDA	1	0	0

Fuente: Autor.

Una vez identificados los elementos del modelo, se procede a insertarlos en la ventana de vista del modelo o vista *3D View* con una acción de arrastrar y soltar (*Drag and Drop*) desde la librería de objetos (*Library*). Una vez insertados, se cambian sus nombres entrando a la ventana de propiedades del objeto con doble clic o haciendo clic derecho sobre el objeto y seleccionando la opción *Properties*. En la parte superior de esta ventana al lado de la imagen asociada a ese objeto de Flexsim se puede introducir el nombre deseado, siguiendo unas reglas básicas de no comenzar con un número o carácter especial y preferiblemente escribir el nombre en mayúsculas. Una vez editados los nombres el layout debe quedar de la siguiente manera:

Figura 1. Objetos fijos en la vista del modelo



Fuente: Autor.

## CONEXIÓN DE OBJETOS

Una vez agregados los objetos se procede a conectarlos para indicar la dirección en la que fluyen las entidades a través del sistema. Los objetos son conectados entre sí a través de puertos de entrada y de salida, manteniendo presionada la tecla *A* y haciendo clic en el objeto origen y posteriormente en el objeto destino, es decir se conecta la salida de un objeto con la entrada del siguiente.


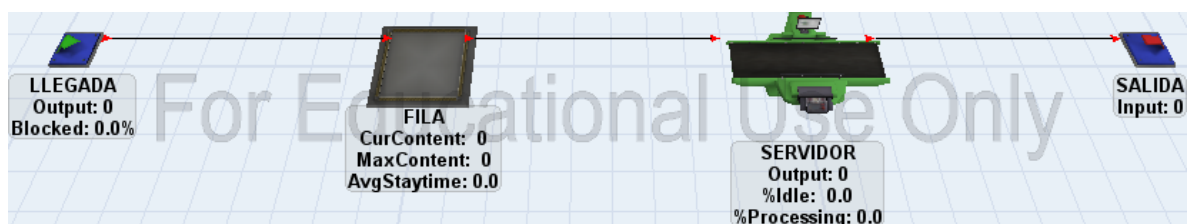
Este tipo de conexiones se realiza manteniendo oprimida la tecla *A* y cliqueando en los dos objetos a conectar. También puede realizarse oprimiendo el siguiente botón  de la barra de herramientas (o *Toolbar*) y posteriormente cliqueando en los objetos a conectar. Es importante el orden de la conexión, ya que esta indica el flujo de entidades y tiene una dirección. El layout del modelo se muestra a continuación:

Figura 2. Objetos fijos conectados



Fuente: Autor.

En Flexsim por defecto cuando se conectan los objetos se puede observar sobre estos sus puertos y la conexión entre ellos a través de una línea negra que los une y que sale de un puerto de color rojo en forma de cono horizontal sobre el objeto y llega a un puerto de entrada de color verde sobre el objeto, tal como se muestra a continuación:

**Figura 3. Conexión de objetos**



Fuente: Autor.

Para información detallada acerca de la conexión de puertos se puede consultar la guía de Flexsim o el instructivo de aprendizaje que acompaña a este manual de prácticas.

## ASIGNACIÓN DE TIEMPOS

En cuanto a los tiempos es necesario establecer si estos son determinísticos, estocásticos, o tienen una distribución empírica asociada (se verá con más detalle en el Taller No. 02) o si se tienen unos tiempos históricos u observados sin ajustar.

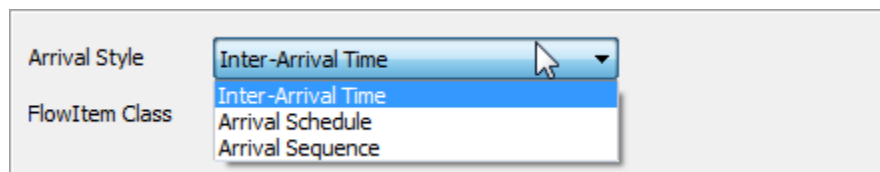
Para este ejercicio guiado, el enunciado dice que la tasa media de llegadas ( $\lambda$ ) es de 20 clientes (o productos) por hora, es decir, que el tiempo entre llegadas ( $1/\lambda$ ) es igual a  $1/20$  clientes por hora o 0.05 horas, pero como se definieron las unidades de tiempo del modelo en minutos, el tiempo entre llegadas es de 3 minutos.

La tasa media de servicio es de 25 clientes (o productos) por hora, es decir, que el tiempo esperado de servicio o proceso ( $1/\mu$ ) es igual a  $1/25$  clientes por hora o 0.04 horas, que equivalen a 2.4 minutos.

Una vez calculados los tiempos, se asigna una distribución estadística de probabilidad que se ajuste al sistema de colas del ejemplo, normalmente se simula la tasa media de llegadas con una distribución Poisson, que indica la probabilidad de que ocurran  $x$  éxitos por unidad de tiempo, área, producto, pieza, etc., pero como en Flexsim se debe alimentar el tiempo entre llegadas que está dado en unidades de tiempo, habitualmente se toma la inversa de la Poisson y se simula con una distribución Exponencial por la relación existente entre la distribución exponencial y el proceso Poisson (con frecuencia llamada exponencial negativa).

De las figuras 4 a la 9 se ilustra la configuración del tiempo entre llegadas y del tiempo de servicio en Flexsim, tema también explicado en el instructivo de aprendizaje en el subtítulo 4.1.

**Figura 4. Tipos de llegada de las entidades**



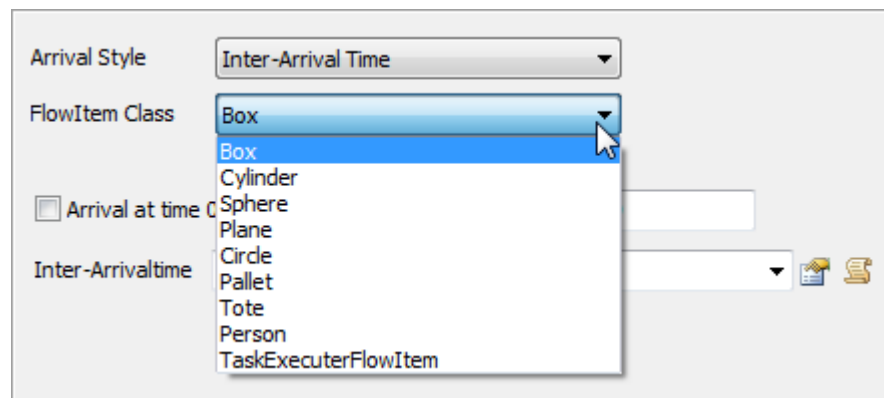
Fuente: Interfaz de Flexsim.

En la figura 4 se visualizan las opciones disponibles para definir el tipo de llegada (3 opciones disponibles: tiempo entre arribos o llegadas, programación de llegada y secuencia de llegada) que para el ejercicio es tiempo entre llegadas, por lo tanto en *Arrival Style* se escoge la opción que trae por defecto: **Inter-Arrival Time**.

En la figura 5 se escoge el tipo de flowitem a crear, que para este ejemplo puede ser *Person* o *Box* para simular personas o productos respectivamente.

Todos los diferentes tipos de flowitem tienen en común la pestaña de propiedades y de etiquetas (Labels), y se diferencian únicamente en el gráfico asociado que tienen, a excepción del *TaskExecuterFlowitem* que es una clase especial.

**Figura 5. Clases de FlowItems**



Fuente: Interfaz de Flexsim.

Figura 6. Tiempo entre llegadas

Source Flow Triggers Labels General Statistics

Arrival Style Inter-Arrival Time  $1/\lambda$

FlowItem Class Box

Arrival at time 0 Item Type 1.00

Inter-Arrivaltime exponential(0, 3, 0)

Tiempo entre llegadas:  $1/\lambda$

Tasa media de llegadas:  $\lambda$

$\lambda = 20$  clientes por hora

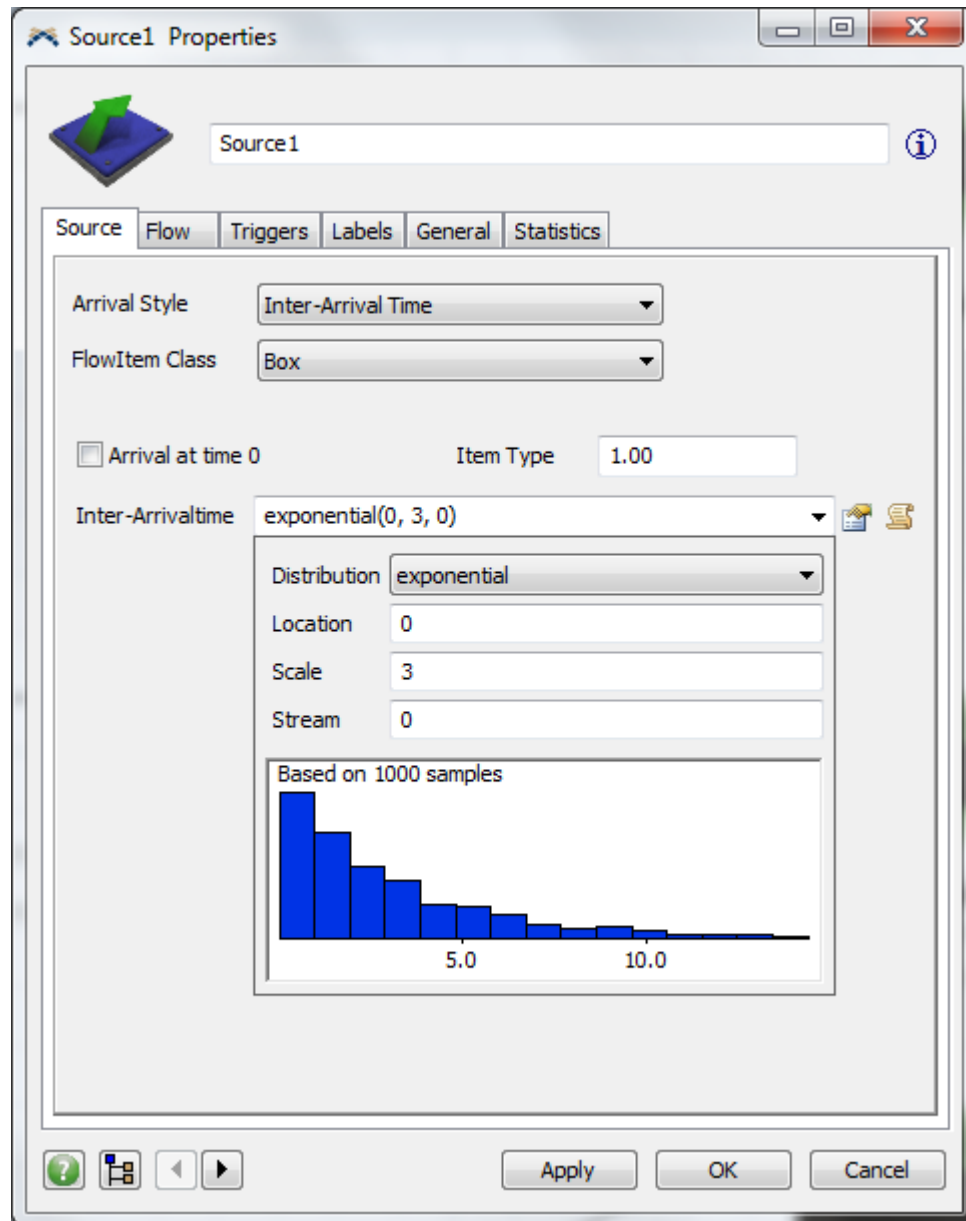
$1/\lambda = 1/20 = 0.05$  horas o 3 minutos

Fuente: Autor.

Hay que tener en cuenta si se desea que la primera entidad o flowitem llegue al sistema en el primer instante de la simulación o minuto 0, para lo cual se selecciona la opción **Arrival at time 0**.

Además es importante no confundir la tasa media de llegadas con el tiempo entre llegadas, que son conceptos diferentes.

Figura 7. Asignación del tiempo entre llegadas



Fuente: Interfaz de Flexsim.

La distribución exponencial necesita como parámetro la media, que se introduce en el segundo parámetro denominado "Scale" de la distribución exponencial en Flexsim.

Figura 8. Tiempo de operación

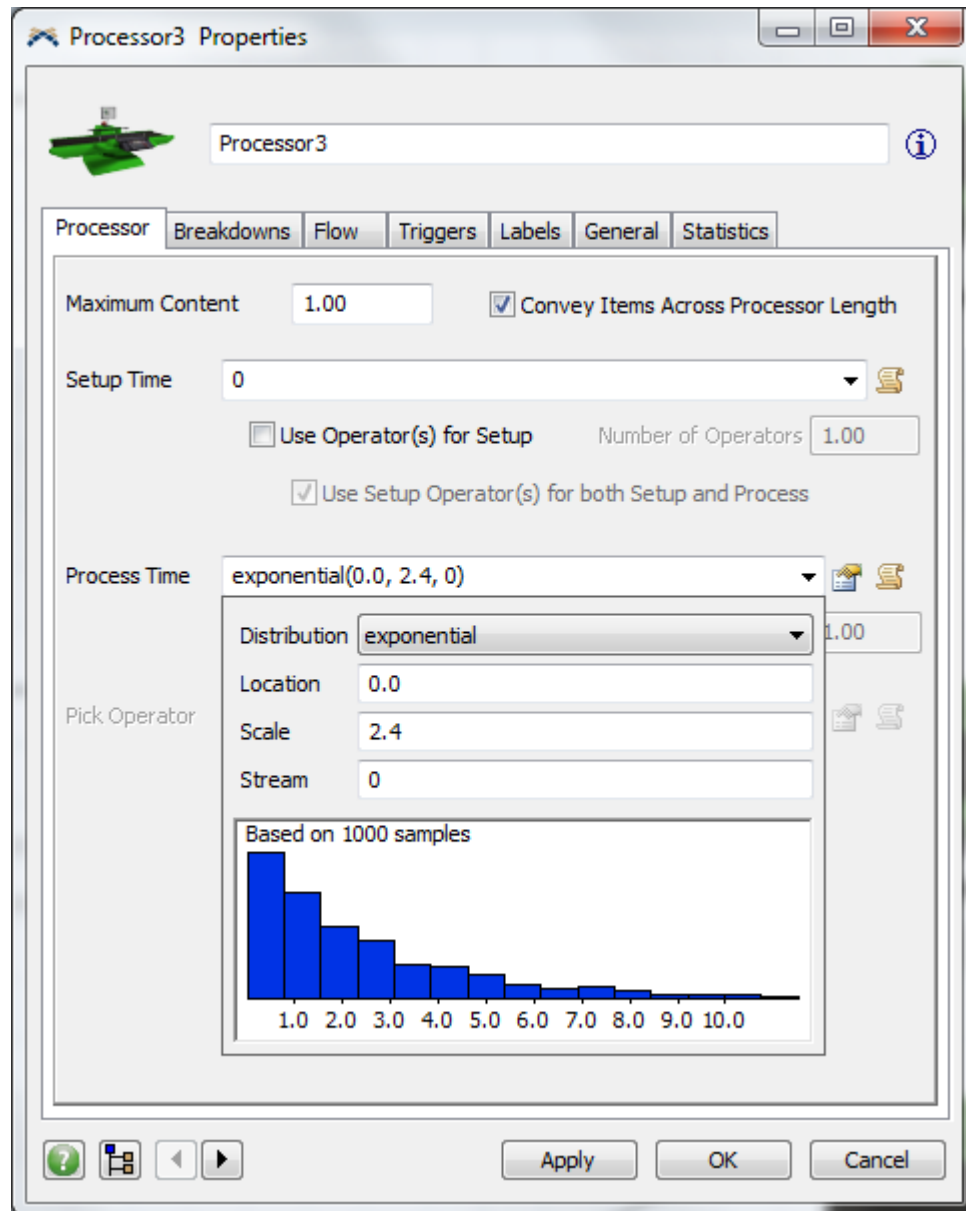
The screenshot shows the 'Processor' configuration window in Flexsim. The 'Process Time' is set to an exponential distribution with parameters (0.0, 2.4, 0). The 'Number of Operators' is set to 1.00. The 'Pick Operator' is set to 'centerobject(current, 1)'. The 'Priority' is 0.00 and 'Preemption' is 'no preempt'. The window also displays the following text:

**Tiempo esperado de servicio:  $1/\mu$**   
**Tasa media de servicio:  $\mu$**   
 $\mu = 25$  clientes por hora  
 $1/\mu = 1/25 = 0.04$  horas o 2.4 minutos

Fuente: Autor.

El tiempo de servicio sigue una distribución exponencial que requiere de la media y que se introduce en Flexsim como el segundo parámetro de dicha distribución en Flexsim.

Figura 9. Asignación del tiempo de operación



Fuente: Interfaz de Flexsim.

Si se desean tiempos determinísticos simplemente en el cuadro de texto correspondiente a *Process Time* se escribe directamente el valor, de igual manera, se haría con el tiempo entre llegadas en el objeto *Source*.

## ATRIBUTOS Y VARIABLES

Es necesario definir 3 variables globales para hacer el cálculo del tiempo de flujo promedio y 3 etiquetas en las entidades o flowitems para capturar la llegada de la entidad al sistema, su tiempo de ciclo y el número de entidad correspondiente. Las etiquetas de las entidades se agregan en el *Flowitem Bin* en el menú Tools y se les asignan valores en los *Triggers* o controladores de los objetos fijos.

**Tabla 2. Etiquetas de los objetos del modelo**

Objeto	Etiqueta	Descripción
<b>Flowitem Person</b>	Entidad	Número de Entidad
	TLlegada	Tiempo de Llegada de la Entidad
	TC	Tiempo de Ciclo para el Flowitem
<b>Source LLEGADA</b>	Entidad	Número de Entidad
	TLlegada	Tiempo de Llegada o de Arribo

Fuente: Autor.

**Tabla 3. Variables globales del modelo**

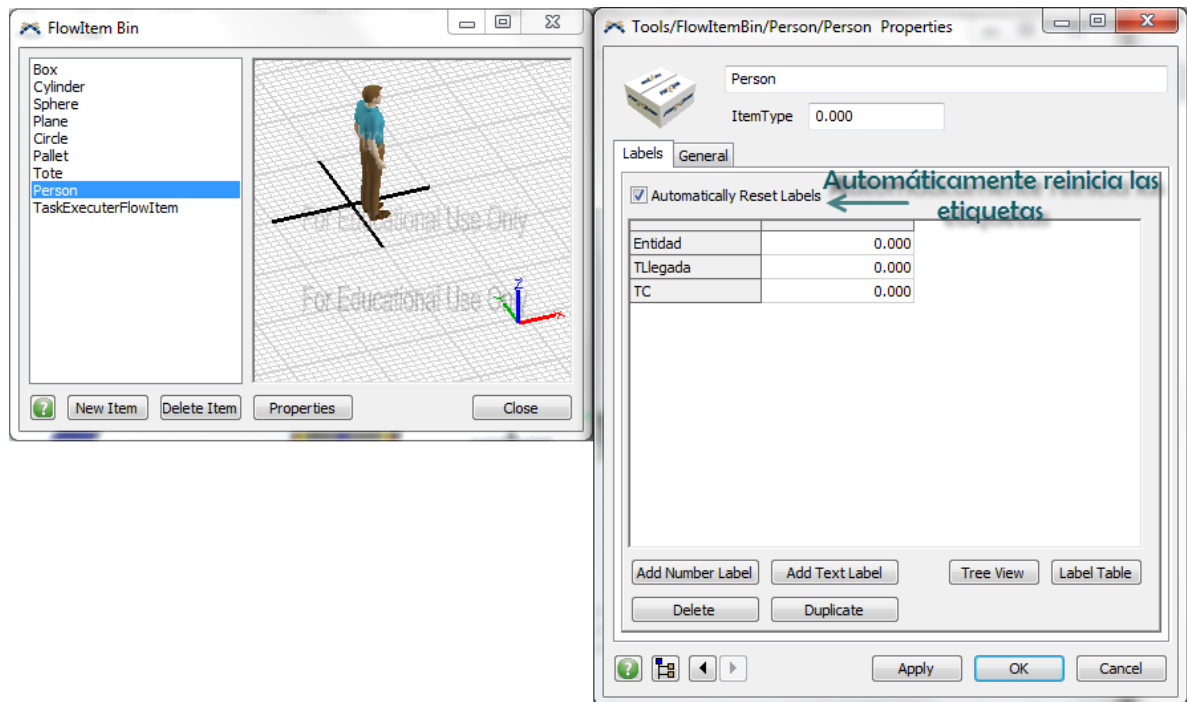
Variable Global	Tipo de variable	Descripción
<b>TSi</b>	Double	Tiempo en el Sistema de la Entidad i
<b>SumTC</b>	Double	Sumatoria del Tiempo de Ciclo
<b>MFT</b>	Double	Tiempo de Flujo Promedio

Fuente: Autor.

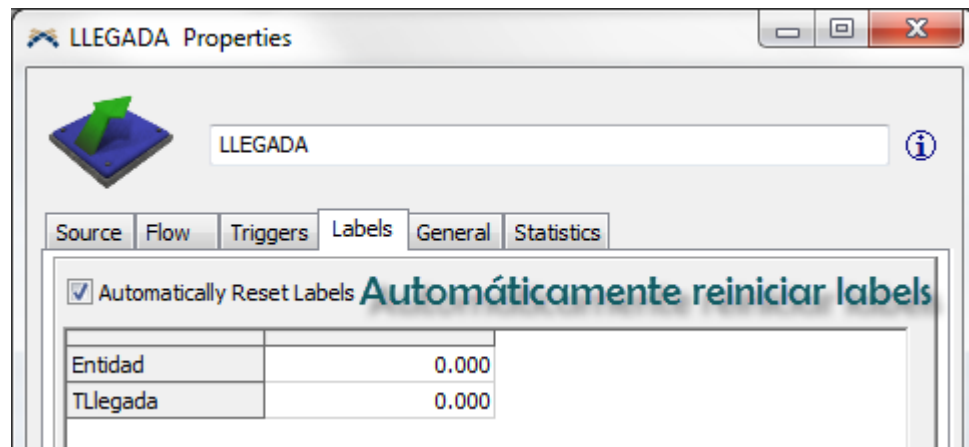
Es importante inicializar las variables globales y las etiquetas a cero, para que no acumulen valores entre réplicas y afecten las estadísticas del modelo.

La figura 10 ilustra donde se crean las etiquetas de las entidades y como inicializarlas cuando se reinicie el modelo. En la pestaña *Labels* de todos los objetos (incluidos los flowitems), se crean las etiquetas o atributos.

**Figura 10. Labels del Flowitem Person**

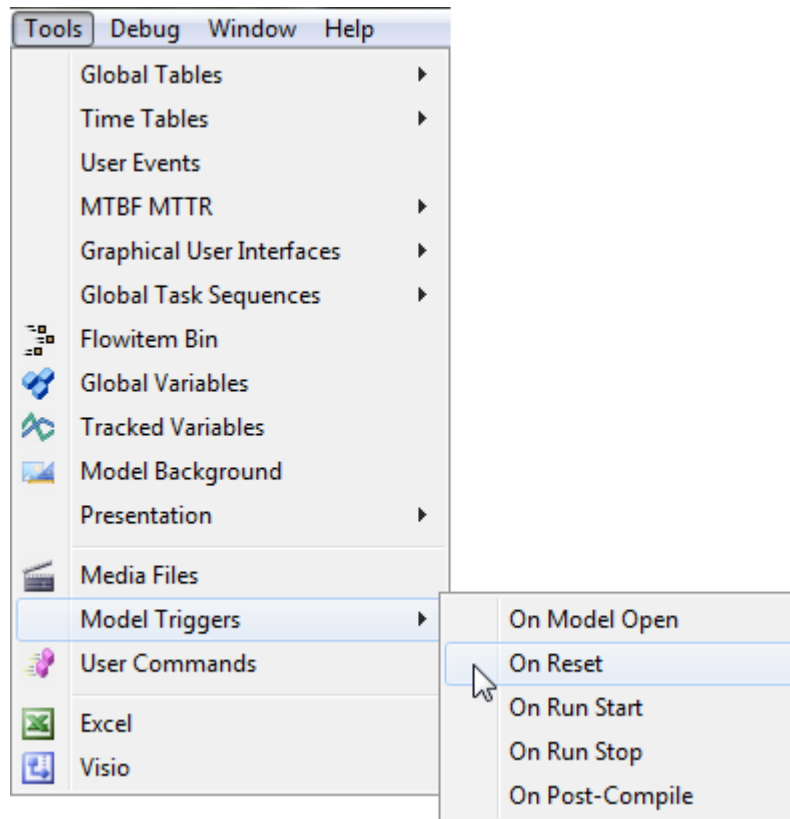


**Figura 11. Labels del Source LLEGADA**



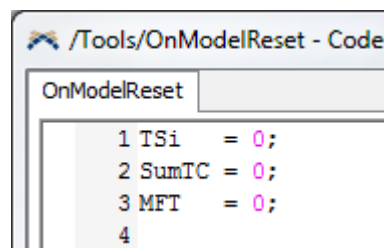
La inicialización de las variables se realiza en los *Triggers* del modelo, específicamente en el *Trigger OnReset*.

**Figura 12. Trigger OnReset del modelo**



Se asigna un valor de cero a todas las variables globales del modelo.

**Figura 13. Trigger OnReset del modelo**



## CONFIGURACIÓN DE TRIGGERS

Los *Triggers* son controladores o disparadores de eventos de los objetos en Flexsim. Se encuentran en la pestaña *Triggers* de las propiedades del objeto y algunos en la pestaña general o principal de objetos fijos en Flexsim.

Un ejemplo de *trigger* común a todos los objetos es *OnReset* que se ejecuta cuando se resetea o reinicia el modelo. *OnEntry* cuando entra una entidad al objeto, lo mismo cuando sale la entidad con el trigger *OnExit*, que son los controladores comunes a todos los objetos. Hay algunos específicos para cada objeto. Para el ejercicio solamente serán necesarios los controladores *OnEntry* y *OnExit*:

### Estructura del Trigger OnEntry / OnExit:

- ❖ **Entry Trigger:** Esta función se ejecuta cada vez que un flowitem entra en el objeto.
- ❖ **Exit Trigger:** Esta función se ejecuta cada vez que un flowitem sale del objeto.

### Variables de Acceso:

- ❖ **current:** el objeto actual
- ❖ **item:** el flowitem que acaba de entrar/salir.
- ❖ **port:** el número del puerto a través del cual el flowitem entró al objeto.

Todos los controladores o *triggers* tienen un código de programación predeterminado, dependiendo de la acción a ejecutar en el controlador, lo que hace más fácil la edición de código.

Como norma se acostumbra a comentar líneas de código de la siguiente manera:

```
inc(label(current,"ClientesCola"),1); //Incrementar en 1 la etiqueta ClientesCola
```

El comentario está después del doble // en color verde, quiere decir que Flexsim no compilará el texto a la derecha. Otra forma de comentario es el que está comprendido entre los siguientes símbolos:

```
/*  
Esto es un comentario  
*/
```

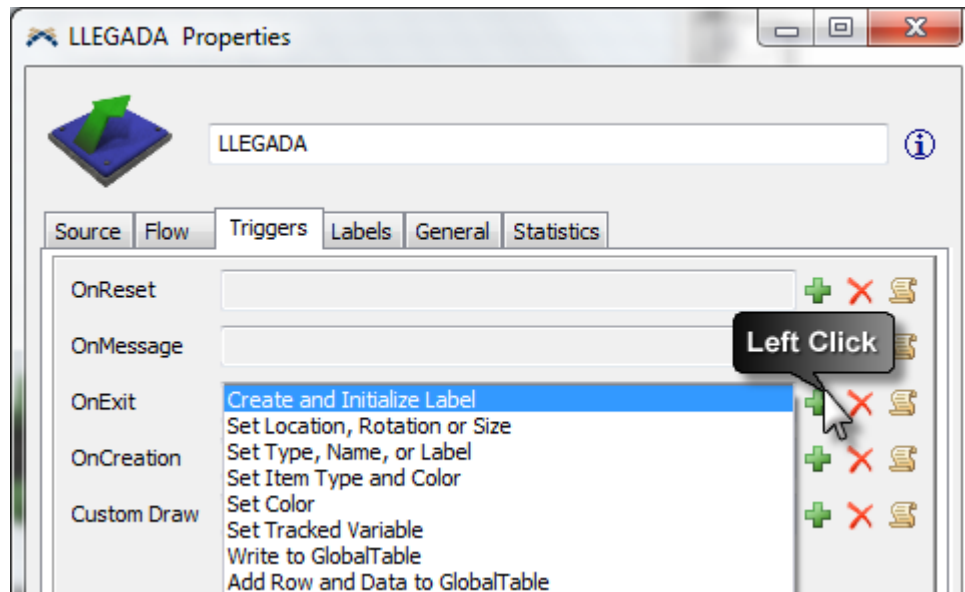
En los flowitems es necesario capturar el tiempo de llegada de los flowitems para hallar el tiempo que permanece la entidad en el sistema, llamado tiempo de ciclo, que es la diferencia entre su salida y su llegada, por lo tanto, se emplean las etiquetas *TLlegada* y *TC* respectivamente. Aunque no es necesario guardar el número de entidad del sistema, se hace útil para hacer el seguimiento de las mismas entidades.

El tiempo actual de simulación se captura con la función `time()` y la asignación de un valor a una etiqueta se hace con la instrucción:

```
setlabelnum(objeto, etiqueta, valor)
```

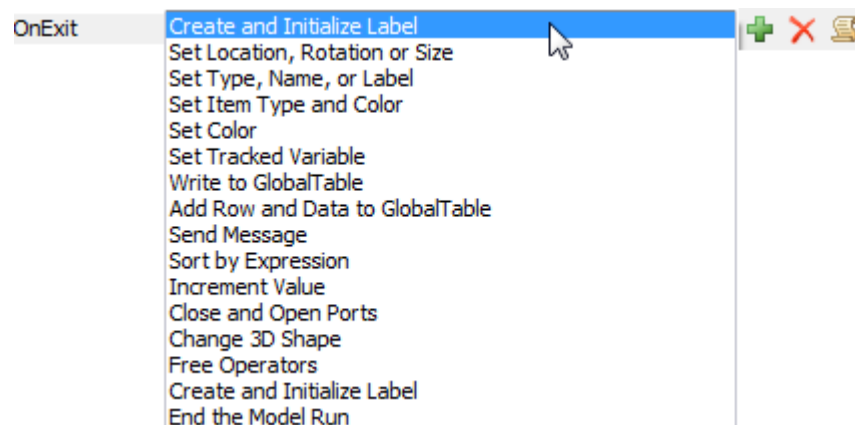
Esa instrucción asigna un `valor` numérico a una `etiqueta` del `objeto` pasado como primer parámetro. En este caso se desea guardar el valor del tiempo de llegada al sistema o de salida del *Source* LLEGADA en la etiqueta *TLlegada* del objeto item, que hace referencia a los flowitems que son creados en el *Source*. En las siguientes figuras se muestran los pasos a seguir en los *Triggers* mencionados.

Figura 14. Trigger OnExit

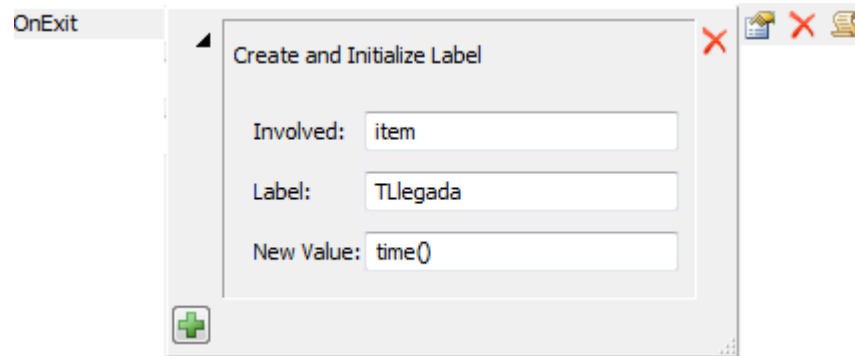



Se presiona en el botón para activar la lista desplegable o Pick list del Trigger en cuestión.

Figura 15. Lista desplegable o picklist del Trigger OnExit

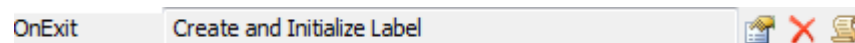



Hay dos maneras de asignar el tiempo en el que se ejecuta el Trigger *OnExit* a la etiqueta *TLlegada* de las entidades o flowitems del sistema, primero con la opción ***Create and Initialize Label*** y segundo con el trigger ***Set Type, Name, or Label***.

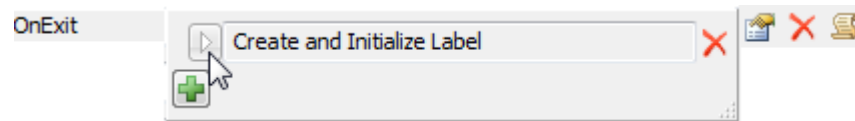
**Figura 16. Ventana emergente del código crear e inicializar etiqueta**

Muchas opciones de la lista desplegable utilizan ventanas emergentes o *popups*. Los *Popups* permiten editar fácilmente los parámetros de la opción escogida en la lista desplegable o *picklist*. Puede modificar estas opciones en cualquier momento haciendo clic en el botón . Una vez que haya introducido los valores que desee en la ventana emergente, haga clic en cualquier lugar fuera de la ventana emergente para cerrarla.

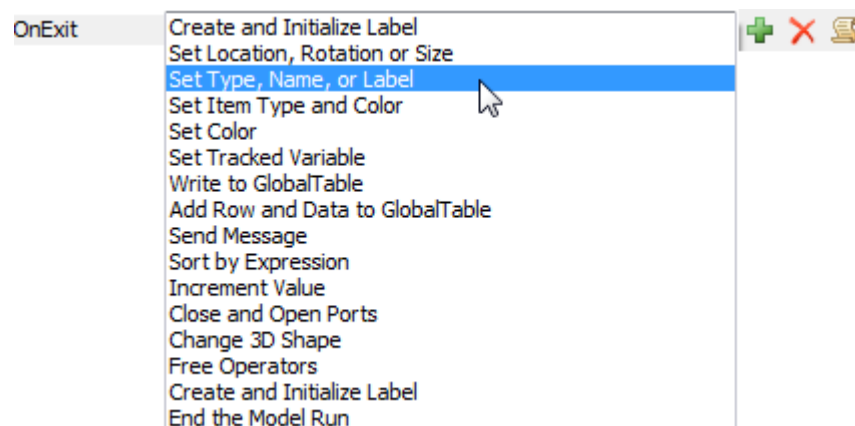
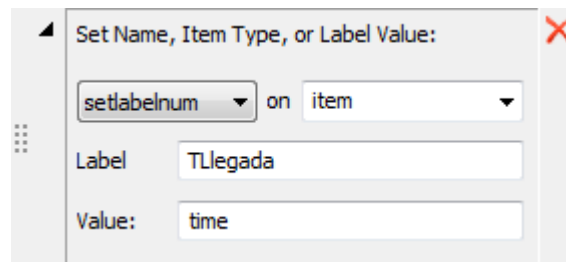
Los parámetros necesarios para asignar el tiempo a la etiqueta *TLlegada* del flowitem, se observan en el *popup* de la figura 16. Una vez introducidos los parámetros, se cliquea afuera del *popup* para cerrarlo, tal como aparece en la figura 17.

**Figura 17. Opción Create and Initialize Label**


Si se desea ingresar nuevamente al *popup* que contiene los parámetros introducidos anteriormente, se presiona el botón , donde aparecen las opciones escogidas de la ventana emergente del *Trigger* como se ilustra en la figura a continuación.

**Figura 18. Picklist escogido en el Trigger**

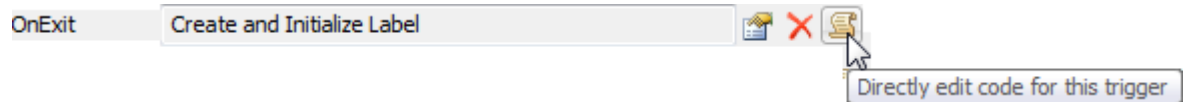
La segunda forma de asignar el tiempo a la etiqueta *TLlegada* se muestra en la figura 19.

**Figura 19. Opción Set Type, Name, or Label****Figura 20. Ventana emergente para la opción Set Label**

En las figuras anteriores se muestra el procedimiento para asignar un valor a una etiqueta mediante la selección de una opción de la lista desplegable del *Trigger* e ingresando los parámetros correspondientes en las ventanas emergentes.

Sin embargo, en algunas ocasiones no basta con emplear los *popups* y editar las plantillas correspondientes, es necesario editar el código predeterminado de la opción escogida en el *Trigger*, para lo cual se presiona el botón  tal como en la figura 21.

**Figura 21. Edición del código fuente de los Triggers**



La figura a continuación tiene el código correspondiente a la primera forma **Create and Initialize Label** descrita anteriormente.

**Figura 22. Código fuente de la opción Create and Initialize Label**

```

1 treenode item = parnode(1);
2 treenode current = ownerobject(c);
3 int port = parval(2);
4
5 //***** PickOption Start *****\\
6 {
7 /**popup:CreateInitializeLabel*/
8 /**Create and Initialize Label*/
9 /** \nObject: */
10 treenode involved = /**tag:involved**/**/item/**/;
11 /** \nLabel: */
12 string labelname = /**tag:labelname**/**/"Llegada"/**/;
13 /** \nValue: */
14 double newvalue = /**tag:newvalue**/**/time()/**/;
15 /**\n\n*/
16 set(assertlabel(involved,labelname, DATATYPE_NUMBER),newvalue);
17 }
18 //***** PickOption End *****\\

```

Todos las opciones de los *Triggers* tienen un código fuente predeterminado asociado, lo cual facilita la programación o edición por parte de los usuarios de Flexsim. A continuación se presenta el código fuente para el *Trigger OnExit* de la fuente LLEGADA.

Figura 23. Código fuente del Trigger OnExit (líneas 1 – 47)

```
1 treenode item = parnode(1);
2 treenode current = ownerobject(c);
3 int rownumber = parval(2); //row number of the schedule/sequence table
4 { //***** PickOption Start *****\
5 /**popup:IncrementValue*/
6 /**Increment Value*/
7 inc(
8 /** \nNode: */ /**tag:node*//**/label(current, "Entidad")/**/
9 ,
10 /** \nIncrement By: */ /**tag:value*//**/1/**/
11 );
12 /**\n*/
13
14 } //***** PickOption End *****\
15 { //***** PickOption Start *****\
16 /**popup:SetCombo*/
17 /**tag:title*//**Set Label Value*/
18 /**tag:command*//**/setlabelnum/**/(
19 /**tag:item*//**/item/**/,
20 /**tag:var*//**/"Entidad"/**/
21 /**tag:value*//**/,getlabelnum(current, "Entidad")/**/
22 );
23 /**\n\n*/
24
25 } //***** PickOption End *****\
26 { //***** PickOption Start *****\
27 /**popup:SetCombo*/
28 /**tag:title*//**Set Label Value*/
29 /**tag:command*//**/setlabelnum/**/(
30 /**tag:item*//**/item/**/,
31 /**tag:var*//**/"TLlegada"/**/
32 /**tag:value*//**/,time/**/
33 );
34 /**\n\n*/
35
36 } //***** PickOption End *****\
37 { //***** PickOption Start *****\
38 /**popup:SetCombo*/
39 /**tag:title*//**Set Label Value*/
40 /**tag:command*//**/setlabelnum/**/(
41 /**tag:item*//**/current/**/,
42 /**tag:var*//**/"TLlegada"/**/
43 /**tag:value*//**/,time/**/
44 );
45 /**\n\n*/
46
47 } //***** PickOption End *****\
```

**Figura 24. Código fuente del Trigger OnExit (líneas 48 – 70)**

```
48 { //***** PickOption Start *****\\
49 /**popup:SetTypeAndColor*/
50 /**Set Itemtype and Color*/
51 /** \nFlowitem: */
52 treenode involved = /**tag:involved/**/item/**/;
53 /** \nItemtype: */
54 double newtype = /**tag:type/**/getlabelnum(item, "Entidad")/**/;
55 /**\n\n*/
56 setitemtype(involved,newtype);
57 colorarray(involved,newtype);
58
59 } //***** PickOption End *****\\
60 { //***** PickOption Start *****\\
61 /** \nSet Name\n*/
62 /** \nvalor: */
63 int    valor = /**/trunc(getlabelnum(item, "Entidad"))/**/;
64 /** \ntexto: */
65 string texto = /**/numtostring(valor, 0, 0)/**/;
66 /** \nComando: */
67 /**/setname(item,concat("Entidad", texto));/**/
68 /**\n*/
69
70 } //***** PickOption End *****\\
```

Dado que para este ejemplo todas las entidades no deben tener el mismo nombre como sucede por defecto en Flexsim, sino un texto y un número consecutivo, se debe asignar un nombre a las entidades, lo cual se puede hacer con la opción *Set Type, Name, or Label* (el código generado por el popup de la figura 25, ya está incluido y editado en el código fuente de la figura 24)


**Figura 25. Opción Set Type, Name, or Label**

Set Name, Item Type, or Label Value:

setname on item

Name: Entidad

Value:

Con la anterior opción se asigna un texto como nombre del flowitem, pero no funciona para adicionarle a ese nombre un número consecutivo, por lo cual, se hace necesario editar el código predeterminado de esa opción haciendo click en  y cambiando el código de la figura 26 por el de la figura 27 que es el correspondiente al de las líneas 60 a 70 de la figura 24.


### Figura 26. Código original a editar

```
{ //***** PickOption Start *****\\
/**popup:SetCombo*/
/**tag:title*//**Set Name*/
/**tag:command*/**/setname/**/ {
/**tag:item*/**/item/**/,
/**tag:var*/**/"Entidad"*/**/
/**tag:value*/**/***/
};
/**\n\n*/
} //***** PickOption End *****\\
```

### Figura 27. Código editado

```
{ //***** PickOption Start *****\\
/** \nSet Name\n*/
/** \nvalor: */
int valor = /**/trunc(getlabelnum(item, "Entidad"))/**/;
/** \ntexto: */
string texto = /**/numtostring(valor, 0, 0)***/;
/** \nComando: */
/**/setname(item,concat("Entidad", texto));/**/
/**\n*/
} //***** PickOption End *****\\
```

Una vez agregadas las instrucciones para capturar el tiempo de llegada de las entidades, solo queda capturar el tiempo de salida y restarle el tiempo de llegada para hallar el tiempo que permaneció en el sistema la entidad. Esta captura siempre se debe realizar en el *Trigger OnEntry* de la última estación que por lo general es un *Sink*, un *Rack*, o un *Queue*, en este caso el *queue SALIDA*.

Este cálculo debe realizarse de manera manual, dado que Flexsim no tiene un código base para hallar el tiempo en el sistema de las entidades, ni el tiempo de flujo promedio, por lo tanto, como no es necesario ingresar a ningún *picklist*, se cliquee directamente sobre el botón de edición de código  del *Trigger OnEntry* de SALIDA.

**Figura 28. Código personalizado en el Trigger OnEntry de SALIDA**

```
1 /**Custom Code*/
2 treenode item = parnode(1);
3 treenode current = ownerobject(c);
4 int port = parval(2);
5 //Operaciones
6 TSi = time - getlabelnum(item, "TLlegada");
7 SumTC += TSi;
8 MFT = SumTC / content(current);
9 //Guardar en etiquetas
10 setlabelnum(item, "TC", TSi);
11 setlabelnum(current, "MFT", MFT);
12
```

Para que haya mayor claridad en el código fuente, se agregan algunos comentarios que no son compilados.

**Figura 29. Código del Trigger OnEntry con comentarios**

```
1 /**Custom Code*/
2 treenode item = parnode(1); //Flowitem que entra al objeto SALIDA
3 treenode current = ownerobject(c); //Objeto actual: SALIDA
4 int port = parval(2); //Puerto por el que ingresa el item
5 //Operaciones
6 TSi = time - getlabelnum(item, "TLlegada"); //Tiempo en el sistema
7 SumTC += TSi; //Tiempo en el sistema acumulado
8 MFT = SumTC / content(current); //Tiempo medio de flujo
9 //Guardar en etiquetas
10 setlabelnum(item, "TC", TSi); //Guarda tiempo de ciclo en el item
11 setlabelnum(current, "MFT", MFT); //Guarda MFT en el objeto actual
12
```

El código guarda en la etiqueta "TC" de la entidad, el tiempo de ciclo como la operación de capturar el tiempo en que entra al objeto SALIDA menos el tiempo que tenía almacenado, que es el de entrada al sistema, operación realizada con las siguientes instrucciones: `TSi = time - getlabelnum(item,"TLlegada");` y `setlabelnum(item,"TC",TSi);` en las líneas 6 y 10.

Las líneas 7 y 8 calculan el tiempo de ciclo acumulado y el tiempo promedio en el sistema, totalizando los tiempos de ciclo `SumIC += TSi;` y dividiendo por el número de entidades que salen del sistema o que entran a SALIDA: `content(current)`

## EJECUCIÓN DE LA SIMULACIÓN


Una vez realizados los pasos anteriores se corre la simulación oprimiendo el botón *Run* de la barra de herramientas y oprimiendo *Stop* para detenerla (ver figura 30). El tiempo de corrida de simulación se configura en el cuadro de texto *Stop Time*, oprimiendo el botón  y escribiendo la duración de la simulación en las unidades de tiempo establecidas al inicio del modelo, en este caso, minutos. Para el ejemplo se desea correr la simulación por un periodo de 480 minutos (ver figura 31).

Figura 30. Configuración del tiempo de corrida de simulación

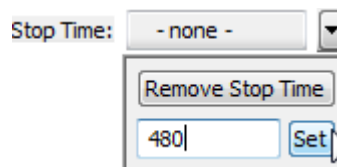
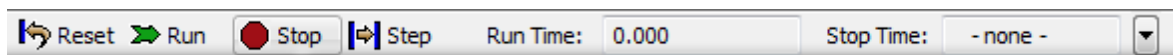


Figura 31. Panel de ejecución de la simulación



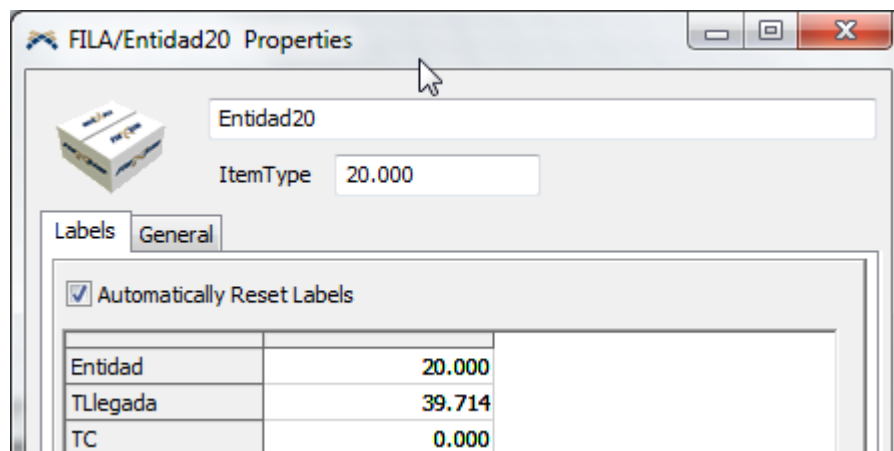
Se corre la simulación y en un instante cualquiera se detiene para observar el comportamiento del sistema, por ejemplo, la simulación fue detenida en el tiempo 45.240 minutos, en el que se observa la siguiente figura.

**Figura 32. Vista del modelo en ejecución**



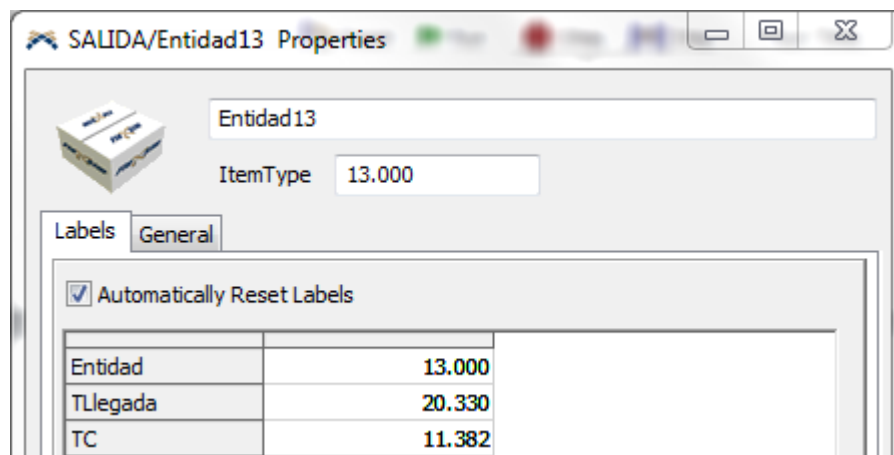
Es posible acceder a las propiedades de las entidades que aún permanecen en el sistema (no han llegado a salida y no tienen tiempo de ciclo), en todo momento de la simulación presionando doble clic sobre la entidad en cuestión. Si la disposición de entidades se hace con un *Sink* no hay manera de acceder al flowitem una vez que este es destruido, por este motivo algunas veces se emplea otro objeto diferente para simular la salida del sistema, tal como un *Queue* o un *Rack*, pero esta solución conlleva a una mayor cantidad de recursos del computador.

**Figura 33. Propiedades de la Entidad20**



Como se configuraron etiquetas en los flowitems y estos no se destruyen sino que permanecen en una cola SALIDA, es posible mirar las etiquetas de los clientes que han salido del sistema, haciendo doble clic en los flowitems. Si se hubiese utilizado un *Sink* en vez del *Queue* no hubiese sido posible entrar a las propiedades de los flowitems una vez que abandonan el sistema, por lo cual se empleó una cola y se guardó en etiquetas de los flowitems su número de entidad, tiempo de llegada y tiempo de ciclo, tal como se observa en la figura 33.

**Figura 34. Propiedades de la Entidad13**



Para determinar el tiempo medio de flujo, no era necesario emplear las 3 etiquetas en los flowitems, únicamente *TLlegada*, las otras dos se utilizaron simplemente para que al acceder a las propiedades de un flowitem en particular, mostrará en sus etiquetas información adicional tal como el tiempo de ciclo y el número de entidad correspondiente, porque con las variables solo se calcula un tiempo promedio de permanencia en el sistema y no hay manera de saber cuál fue el tiempo empleado por una entidad cualquiera, a menos que se guarde ese valor o en una etiqueta del flowitem, como se realizó en este ejemplo, o en una tabla global, tema que será abordado en el siguiente taller.

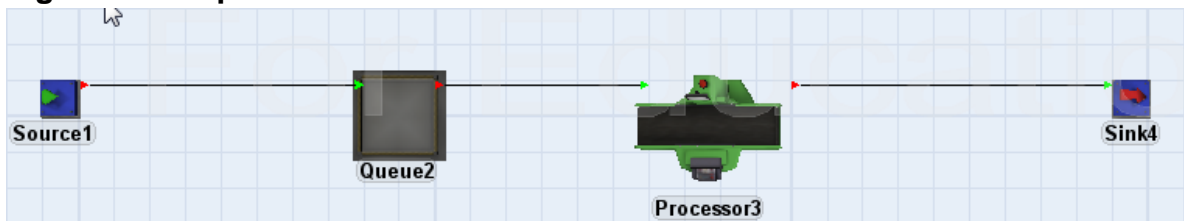
## TALLER A DESARROLLAR

### OBJETOS DEL MODELO

Tabla 4. Objetos del Taller No. 1

Elemento	Objeto	Nombre	Puertos	
			Entrada	Salida
Llegadas	Source	PROGRAMAS	0	1
		CLIENTES	0	1
Operaciones	Processor	SERVIDOR	1	1
		CAJERO	1	1
Filas	Queue	FILA PROGRAMAS	1	1
		FILA CLIENTES	1	1
Salidas	Sink	SALIDA PROGS	1	0
		SALIDA CLIENTES	1	0

Figura 35. Esquema inicial del Taller No. 01



### DURACIÓN DEL TALLER

La duración estimada del taller es de 2 horas.

## DESCRIPCIÓN DEL TALLER

Simular en FLEXSIM los siguientes ejercicios de líneas de espera y contrastar los resultados teóricos y los simulados:

1. En un servidor de la universidad se envían programas de computador para ser ejecutados. Los programas llegan al servidor con una tasa de 10 por minuto. El tiempo medio de ejecución de cada programa es de 5 segundos y tanto los tiempos entre llegadas como los tiempos de ejecución se distribuyen exponencialmente.
  - a) El factor de utilización del servidor.
  - b) ¿El sistema alcanza el estado estacionario?
  - c) ¿Qué proporción de tiempo está el servidor desocupado?
  - d) ¿Cuál es el tiempo esperado total de salida de un programa?
  
2. La ventanilla de un banco realiza las transacciones en un tiempo medio de 2 minutos. Los clientes llegan con una tasa media de 20 clientes a la hora. Si se supone que las llegadas siguen un proceso de Poisson y el tiempo de servicio es exponencial, determinar
  - a) El factor de utilización del sistema.
  - b) El porcentaje de tiempo en el que el cajero está desocupado.
  - c) El factor de utilización del sistema con dos servidores.
  - d) El tiempo medio de espera de los clientes en la cola.

Tiempo de corrida de simulación para los dos ejercicios: Diez mil (10000) minutos.

## RECOMENDACIONES

- Se recomienda tener claridad en conceptos como la tasa media de llegadas y el tiempo entre llegadas.
- Es conveniente activar la vista de las estadísticas en los objetos.

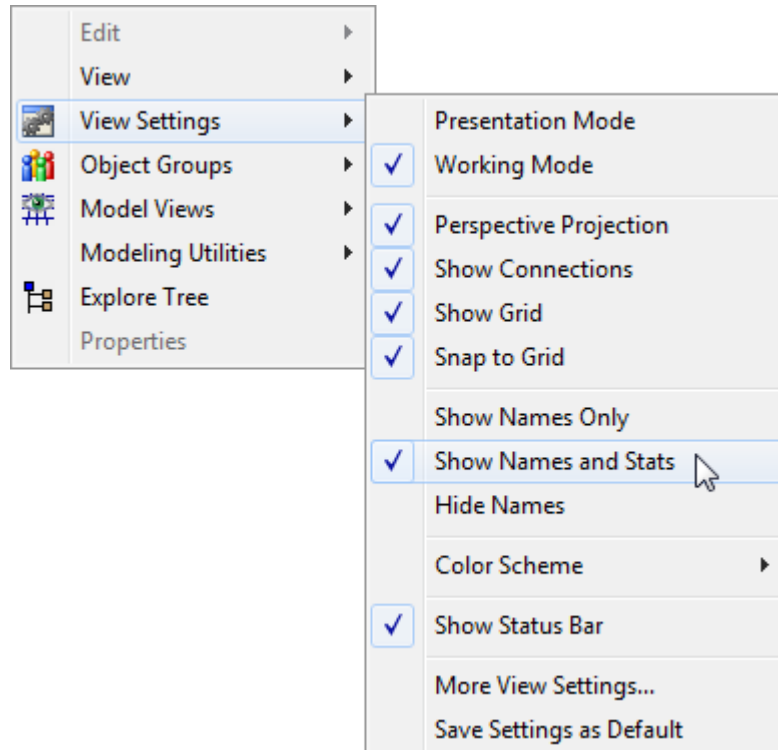


Figura 2. Mostrar nombres y estadísticas.

- Utilice la herramienta *Dashboard* para agregar gráficos que visualicen la información deseada (véase la sección 4.11 del instructivo de aprendizaje).
- Como se usan variables globales para hallar el tiempo de ciclo de los flowitems que circulan en el sistema, es necesario inicializar todas las variables del modelo, para lo cual, se les asigna un valor de cero al reiniciar el modelo o en la opción *Model Triggers -> OnReset*.

## TALLER 2: FILAS DE ESPERA CON TABLAS GLOBALES

### INTRODUCCIÓN

Para la segunda práctica se simulará un sistema similar al del primer taller pero utilizando tablas globales para definir los tiempos de llegada y de servicio y empleando etiquetas en los objetos para determinar estadísticas de interés del modelador o del analista.

Para los inputs o datos de entrada del modelo, se parte de unos tiempos observados que se encuentran en un archivo de Excel adjunto a la práctica, para alimentarlos en Flexsim a través de distribuciones empíricas y tablas globales. En la vida real se ajustan los datos a una distribución de probabilidad estadística, tema que será abordado en la tercera práctica de este manual.

El archivo en Excel contiene relaciones de un sistema de colas básico, en las que las medidas de desempeño son sencillas de calcular pero que en sistemas más complejos y con mayor cantidad de elementos se vuelve más difícil determinar estas fórmulas analíticamente, para lo cual, se suele apelar a un software especializado que simule problemas de líneas de espera o se recurre a la simulación computacional, herramienta que permite hacer cambios muy fácilmente sobre el sistema y hallar fácil o automáticamente algunas medidas de desempeño.

Esos datos observados de entrada que no van a ser ajustados a una distribución estadística pueden ser importados desde Excel, donde posteriormente se pueden exportar los datos de salida que arroje el sistema a una hoja de cálculo.

## OBJETIVOS

- ❖ Desarrollar un modelo de colas en Flexsim mediante la utilización de tablas globales para tiempos entre llegadas y tiempos de operación observados y tiempos que siguen una distribución empírica.
- ❖ Hallar el factor de utilización de los servidores y el tiempo de ciclo del sistema mediante la utilización de variables y etiquetas en los objetos empleados, y compararlos con las estadísticas que arroja Flexsim.

## METODOLOGÍA

La práctica consta de dos partes:

- Parte 1: correspondiente al ejercicio guía, que se acompaña de un archivo en Excel con la simulación en hoja de cálculo (Parte A), un archivo en Flexsim con el desarrollo del ejemplo y un vídeo explicativo de la construcción de algunas partes del modelo (Parte B).
- Parte 2: correspondiente a los ejercicios que el estudiante debe desarrollar.

Para el ejercicio guía se simula un problema de teoría de colas en hoja de cálculo, en este caso, Excel, para implementarlo posteriormente en Flexsim. Se modela un sistema donde se simulan el ingreso de 20 entidades a un sistema de líneas de espera, con unos tiempos entre llegadas y tiempos de servicio que siguen una distribución empírica, es decir, que no se han ajustado a una distribución de probabilidad teórica conocida, sus valores son obtenidos a base de observaciones de las cuales se obtienen sus probabilidades de ocurrencia.

Para generar el tiempo correspondiente se emplean números aleatorios que para efectos de la práctica serán los mismos para todos los estudiantes, con el fin de obtener los mismos resultados en Excel y posteriormente utilizar estos mismos tiempos en Flexsim para comparar la simulación hecha en hoja de cálculo y la realizada en el software de simulación. Una vez contrastados los resultados de la simulación obtenidos por los dos métodos, se implementará el uso de distribuciones empíricas en el ejercicio.

## 1. EJERCICIO GUÍA

### ENUNCIADO

Ejemplo 2.1 extraído de Banks et al. (2001):

Una pequeña tienda de comestibles tiene una sola caja. Los clientes llegan a esta caja aleatoriamente de 1 a 8 minutos de diferencia. Cada posible valor de tiempo entre llegadas tiene la misma probabilidad de ocurrencia, como se muestra en la Tabla 1.

Los tiempos de servicio varían entre 1 y 6 minutos con las probabilidades que se muestran en la Tabla 2. Analizar el sistema simulando la llegada y el servicio de 20 clientes.

**Tabla 1. Distribución de Tiempo entre Arribos**

Tiempo entre Llegadas (Minutos)	Probabilidad	Probabilidad Acumulada	Números Aleatorios Asignados
1	0.125	0.125	001 – 125
2	0.125	0.250	126 – 250
3	0.125	0.375	251 – 375
4	0.125	0.500	376 – 500
5	0.125	0.625	501 – 625
6	0.125	0.750	626 – 750
7	0.125	0.875	751 – 875
8	0.125	1.000	876 – 000

**Tabla 2. Distribución de Tiempo de Servicio**

Tiempo entre Servicio (Minutos)	Probabilidad	Probabilidad Acumulada	Números Aleatorios Asignados
1	0.10	0.10	01 – 10
2	0.20	0.30	12 – 30
3	0.30	0.60	31 – 60
4	0.25	0.85	61 – 85
5	0.10	0.95	86 – 95
6	0.05	1.00	96 – 00

Para la simulación en hoja de cálculo se emplean los siguientes números aleatorios, para tener uniformidad en los resultados de la simulación:

**Tabla 3. Números aleatorios para el ejercicio**

Entidad Número	Aleatorio T. entre Arribos	Aleatorio Tiempo de Servicio
1		84
2	913	10
3	727	74
4	15	53
5	948	17
6	309	79
7	922	91
8	753	67
9	235	89
10	302	38
11	109	32
12	93	94
13	607	79
14	738	5
15	359	89
16	888	84
17	106	52
18	212	55
19	493	30
20	535	50

Los resultados de la simulación en Excel son mostrados en la siguiente tabla:

**Tabla 4. Resultados de simulación en Excel**

Entidad Número	Aleatorio Tiempo. entre Arribos	Tiempo entre Llegadas	T. de Llegada (Reloj de Simulación)	Tiempo de Inicio del Servicio	Aleatorio Tiempo de Servicio	Tiempo de Servicio	Tiempo de Finalización del Servicio	Tiempo de Espera en Cola	Número de Entidades en la Cola	Tiempo de Ciclo (T. en el Sistema)	Tiempo de Inactividad del Servidor	Factor de Utilización (%)
1		0	0	0	70	4	4	0	0	4	0	100,00%
2	900	8	8	8	5	1	9	0	0	1	4	55,56%
3	650	6	14	14	70	4	18	0	0	4	5	50,00%
4	100	1	15	18	50	3	21	3	1	6	0	57,14%
5	900	8	23	23	20	2	25	0	0	2	2	56,00%
6	300	3	26	26	70	4	30	0	0	4	1	60,00%
7	900	8	34	34	90	5	39	0	0	5	4	58,97%
8	800	7	41	41	70	4	45	0	0	4	2	60,00%
9	200	2	43	45	90	5	50	2	1	7	0	64,00%
10	300	3	46	50	50	3	53	4	1	7	0	66,04%
11	100	1	47	53	50	3	56	6	2	9	0	67,86%
12	100	1	48	56	90	5	61	8	3	13	0	70,49%
13	550	5	53	61	70	4	65	8	2	12	0	72,31%
14	650	6	59	65	5	1	66	6	2	7	0	72,73%
15	300	3	62	66	90	5	71	4	2	9	0	74,65%
16	900	8	70	71	70	4	75	1	1	5	0	76,00%
17	100	1	71	75	50	3	78	4	1	7	0	76,92%
18	200	2	73	78	50	3	81	5	2	8	0	77,78%
19	400	4	77	81	20	2	83	4	2	6	0	78,31%
20	550	5	82	83	50	3	86	1	1	4	0	79,07%
<b>Suma</b>		<b>82</b>				<b>68</b>		<b>56</b>	<b>21</b>	<b>124</b>	<b>18</b>	
<b>Promedio</b>		<b>4,32</b>				<b>3,40</b>		<b>2,80</b>	<b>1,05</b>	<b>6,20</b>		

## A. SIMULACIÓN EN EXCEL

El taller consta de un archivo en Excel con la simulación en hoja de cálculo para el ejercicio guía y un archivo en Flexsim con el desarrollo del ejemplo. El estudiante debe desarrollar los ejercicios de la parte B del presente taller.

1. Para llenar la anterior tabla, es necesario buscar los aleatorios del tiempo entre llegadas y del tiempo de servicio de la tabla 3, en la última columna de las tablas 1 y 2 correspondiente al rango o intervalo de aleatorios que pueden tomar estos tiempos. Se puede realizar de manera manual o empleando la fórmula **BuscarV** de Excel.

Ejemplo: Hallar el tiempo entre llegadas para la entidad 10.

El aleatorio de la entidad número 10 correspondiente al tiempo entre llegadas o arribos (ver tabla 3) es 302. Este número se busca en la tabla de distribución de tiempo entre arribos (ver tabla 1), en el rango o intervalo de aleatorios asignados en la última columna.

El aleatorio 302 está comprendido en el intervalo 251 – 375, que le corresponde el tiempo entre llegadas: 3 minutos, disponible en la primera columna de la tabla 1.

2. Una vez calculado el tiempo entre llegadas y de servicio, se procede a hallar el tiempo de llegada e inicio del servicio.

Para el tiempo de llegada se toman los valores acumulados del tiempo entre llegadas.

Para el tiempo de inicio del servicio se toma el máximo valor entre el tiempo de llegada de la entidad actual y el tiempo de finalización del servicio de la entidad anterior.

3. El tiempo de finalización del servicio corresponde al tiempo de inicio del servicio más el tiempo de servicio.
4. El tiempo de espera en cola corresponde a la diferencia entre el tiempo de inicio del servicio y el tiempo de llegada de la entidad.
5. El tiempo de ciclo corresponde a la suma del tiempo de servicio más el tiempo de espera en cola.
6. Para el número de entidades en cola, se compara el tiempo de llegada de la entidad actual y el tiempo de finalización del servicio de las entidades anteriores, si es menor se suman sus ocurrencias.

Ejemplo: la entidad 4 llega al sistema al minuto 15, este valor se compara con los anteriores tiempos de finalización que son: 4, 9 y 18, encontrándose solamente una ocurrencia (18) donde 15 es menor, por lo tanto se toma el valor de todas las ocurrencias, que en este caso equivale a 1.

7. El factor de utilización en el instante en que una entidad termina su servicio, corresponde a la suma de los tiempos de servicio hasta ese instante dividido por el tiempo de finalización del servicio.
8. Por último se sacan los totales y promedios de:

- Tiempo entre llegadas.
- Tiempo de servicio.
- Tiempo de espera en cola y número de entidades en cola.
- Tiempo de Ciclo.

## B. SIMULACIÓN EN FLEXSIM

Para el desarrollo de la parte B de la práctica se puede consultar el vídeo adjunto al taller.

### OBJETOS DEL MODELO

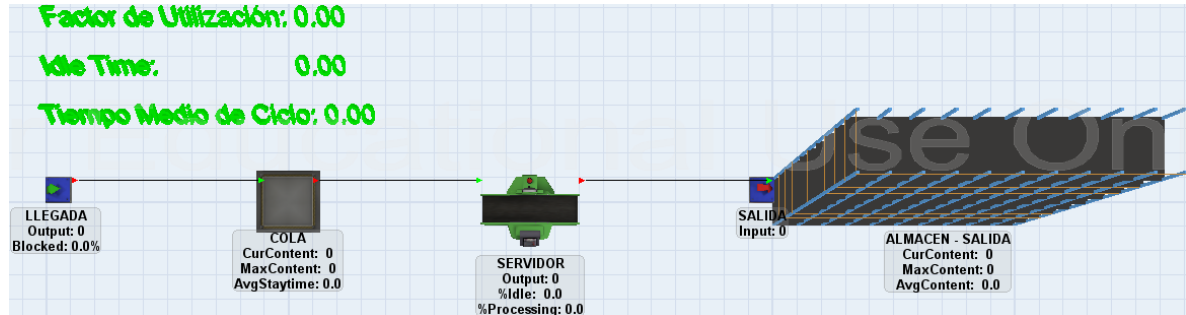
Los objetos requeridos en Flexsim son los siguientes:

**Tabla 5. Objetos del Modelo**

Elemento	Objeto	Nombre	Puertos		
			Entrada	Central	Salida
Llegadas	Source	LLEGADA	0	0	1
Filas	Queue	COLA	1	0	1
Operaciones	Processor	SERVIDOR	1	0	1
Salidas	Sink o Rack	ALMACEN - SALIDA	1	0	0
Texto en 3D	VisualTool	VisualTool1	0	0	0
		VisualTool2	0	0	0
		VisualTool3	0	0	0
		VisualTool4	1	1	1
		VisualTool5	1	1	1
		VisualTool6	1	1	1

El layout del modelo se muestra a continuación:

Figura 1. Esquema inicial del Taller No. 2



### 1. Creación de entidades:

Se realiza con el objeto *Source*. Se agrega desde la librería de objetos arrastrándolo a la ventana de vista del modelo.

### 2. Adición de los objetos fijos del modelo:

Se agregan los demás objetos según el layout de la figura 1 o de la tabla 5, de la misma manera como se agregó el objeto *Source*, es decir, arrastrándolo con clic sostenido desde la biblioteca de objetos hasta la vista 3D del modelo.

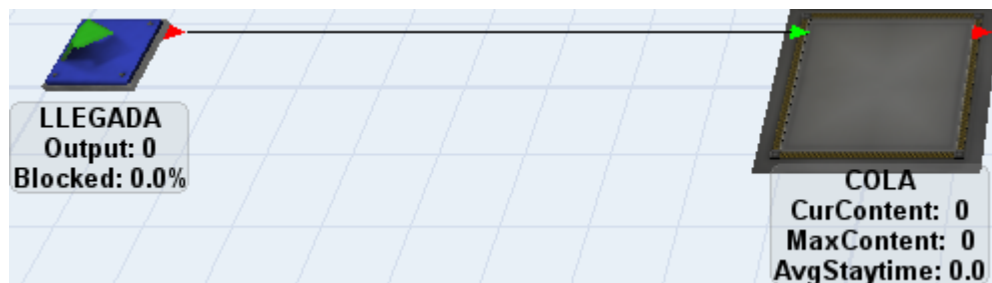
En la figura 1 se observa un objeto *Sink* de nombre SALIDA donde se realizan las salidas de las entidades del sistema, sin embargo para efectos prácticos del taller se utilizará un objeto *Rack* de nombre ALMACEN – SALIDA, donde las entidades no son destruidas sino almacenadas con el fin de poder visualizar las etiquetas o *labels* de las entidades y de paso estudiar la configuración de un *Rack*, por lo tanto se puede omitir este objeto *Sink* del modelo.

### 3. Conexión de objetos:

Los objetos pueden ser conectados entre sí a través de puertos de entrada y salida, manteniendo presionada la tecla *A* y haciendo clic en el objeto origen y posteriormente en el objeto destino, es decir se conecta la salida de un objeto con la entrada del siguiente.

En Flexsim por defecto cuando se conectan los objetos se puede observar sobre estos sus puertos y la conexión entre ellos a través de una línea negra que los une y que sale de un puerto de color rojo en forma de cono horizontal sobre el objeto y llega a un puerto de entrada de color verde sobre el objeto, tal como se muestra a continuación:

**Figura 2. Conexión de objetos**

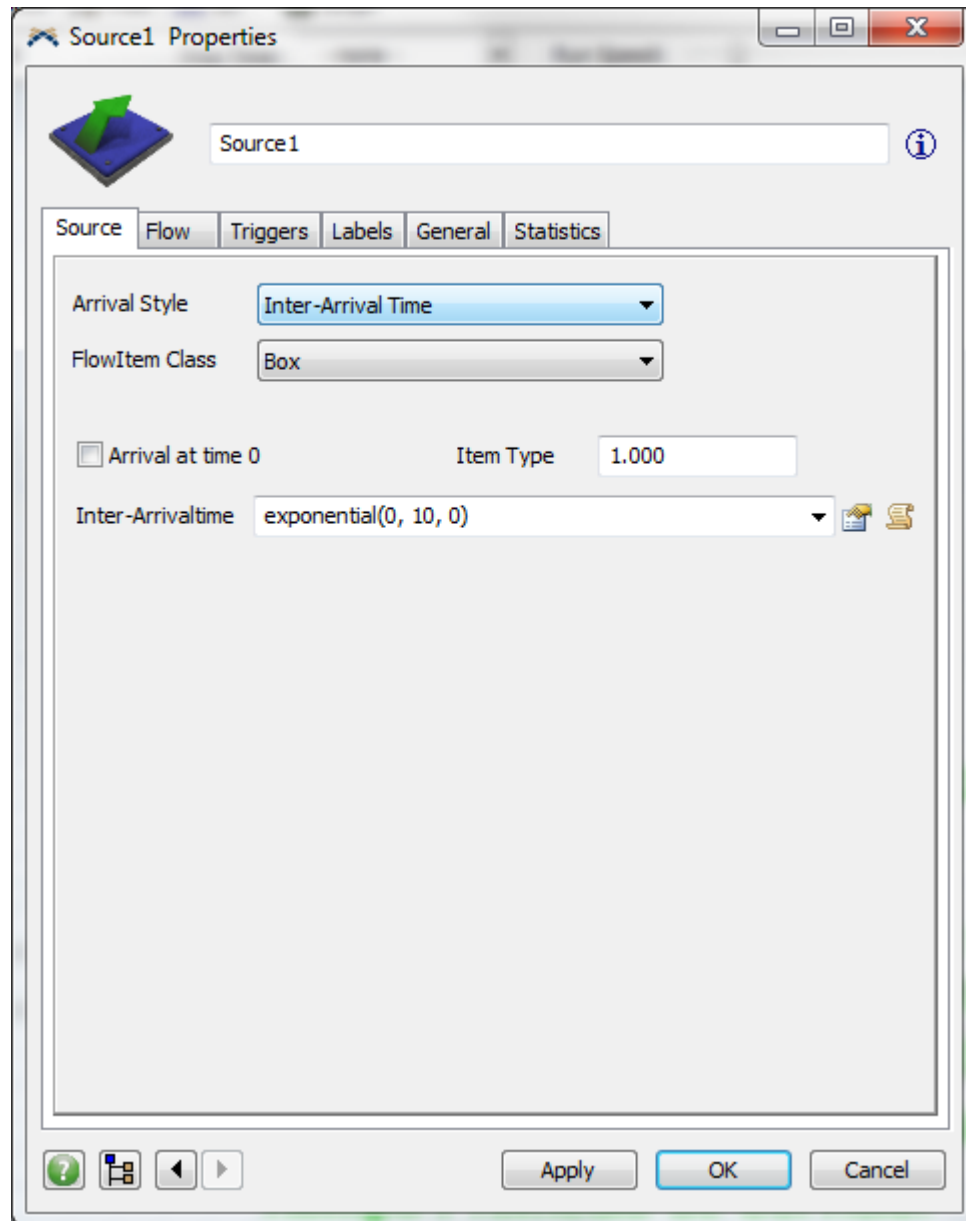


Para información detallada acerca de la conexión de puertos se puede consultar la guía de Flexsim o el instructivo de aprendizaje que acompaña a este manual de prácticas.

### 4. Configuración de las propiedades del objeto Source:

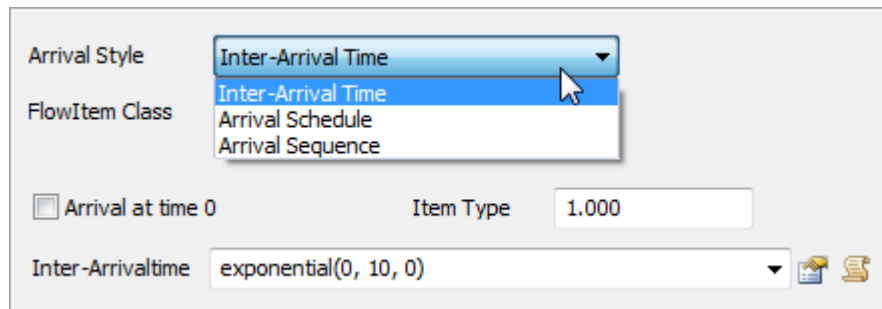
Se hace doble clic sobre Source para que aparezcan las propiedades del objeto, donde se le puede asignar un nombre, elegir el tipo de flowitem a crear y sobre todo la opción más importante: definir el tipo de llegada y el itemtype de los items.

Figura 3. Propiedades del objeto Source

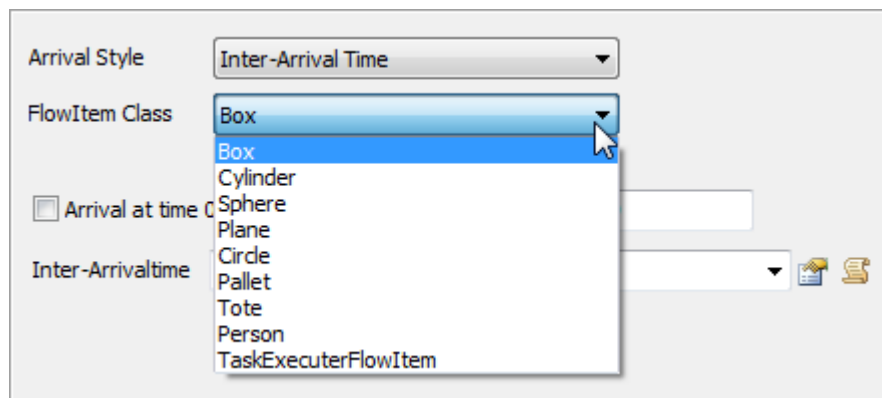


En la figura 3 se observa las propiedades por defecto del objeto Source. En la figura 4 y 5 se visualizan las opciones disponibles para definir el tipo de llegada (tiempo entre arribos, programación de llegada y secuencia de llegada) y el tipo de flowitem a crear respectivamente.

**Figura 4. Tipo de llegada de las entidades**



**Figura 5. Clases de FlowItems**

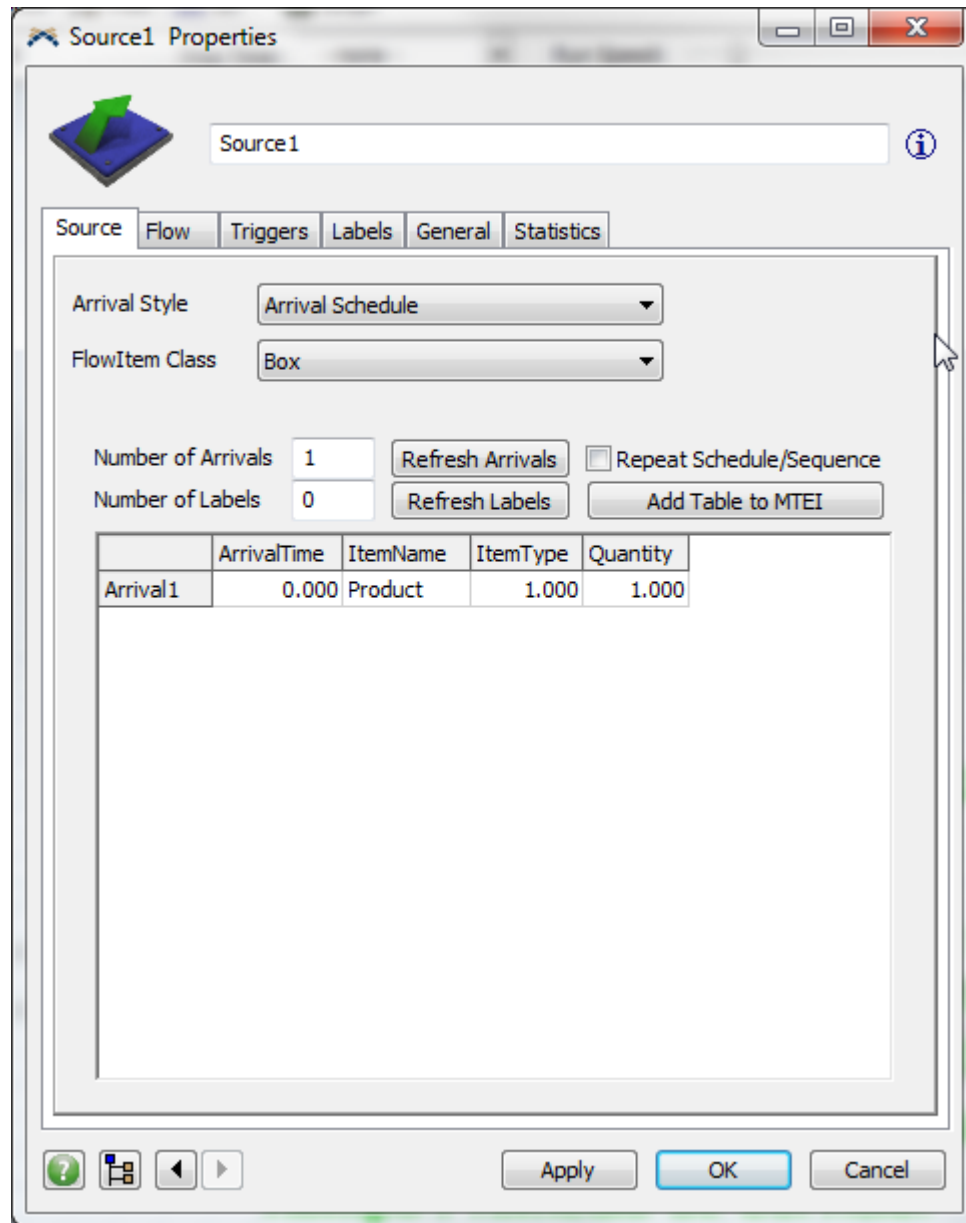


En la figura 6 se observa el estilo de llegada siguiendo una programación o un horario, donde se introducirá el tiempo de llegada (tiempo entre llegadas acumulado) de las 20 entidades del ejercicio, tiempo que se puede ver en la tabla 4. Para este ejercicio no se empleará una distribución de probabilidad, sino una tabla con un tiempo de llegada para cada entidad.

En la figura 7 se encuentra definida la hora de llegada y las etiquetas a emplear en las entidades y no en el objeto LLEGADA. Estas etiquetas son:

- a) Entidad: correspondiente al número de entidad creado.
- b) TCola: etiqueta donde se guardará el tiempo en cola para la entidad.
- c) TCiclo: tiempo de ciclo o en el sistema de la entidad.

Figura 6. Propiedades del objeto Source



Para adicionar etiquetas a las entidades, se puede realizar de tres maneras:

- En el *Flowitem Bin* en el menú Tools.
- En los *Triggers* o controladores del objeto *Source*.

- c) En la tabla que acompaña a la opción *Arrival Schedule* o *Arrival Sequence* en las propiedades del objeto *Source*, opción empleada en este ejercicio. Debe escribir el número de etiquetas o labels y presionar *Refresh Labels* para que aparezcan las etiquetas en la tabla.

**Figura 7. Tiempos de llegada de las entidades**

The screenshot shows the 'LLEGADA Properties' dialog box with the 'Labels' tab selected. The 'Arrival Style' is set to 'Arrival Schedule' and 'FlowItem Class' is 'Box'. The 'Number of Arrivals' is 20 and 'Number of Labels' is 3. The 'Refresh Labels' button is highlighted. Below the settings is a table with the following data:

	ArrivalTime	ItemName	ItemType	Quantity	Entidad	TCola	TCiclo
Arrival1	0.000	Product	1.000	1.000	1.000	0.000	0.000
Arrival2	8.000	Product	1.000	1.000	2.000	0.000	0.000
Arrival3	14.000	Product	1.000	1.000	3.000	0.000	0.000
Arrival4	15.000	Product	1.000	1.000	4.000	0.000	0.000
Arrival5	23.000	Product	1.000	1.000	5.000	0.000	0.000
Arrival6	26.000	Product	1.000	1.000	6.000	0.000	0.000
Arrival7	34.000	Product	1.000	1.000	7.000	0.000	0.000
Arrival8	41.000	Product	1.000	1.000	8.000	0.000	0.000
Arrival9	43.000	Product	1.000	1.000	9.000	0.000	0.000
Arrival10	46.000	Product	1.000	1.000	10.000	0.000	0.000
Arrival11	47.000	Product	1.000	1.000	11.000	0.000	0.000
Arrival12	48.000	Product	1.000	1.000	12.000	0.000	0.000
Arrival13	53.000	Product	1.000	1.000	13.000	0.000	0.000
Arrival14	59.000	Product	1.000	1.000	14.000	0.000	0.000

## 5. Creación de tablas globales para los tiempos de llegada y de servicio:

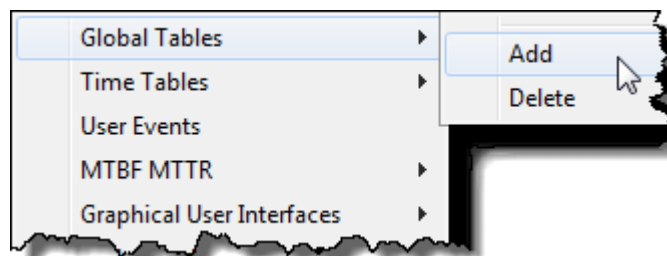
Para la introducción de los tiempos de llegada para cada entidad se puede utilizar una tabla global o la tabla que trae la opción *Arrival Schedule* del objeto *Source*, tal como se ilustra en la figura 7. El uso de la tabla global se dejará para el tiempo de servicio para abordar los dos métodos.

Para definir el número de entradas, se coloca un número en el campo *Number of Arrivals*, de igual manera se hace con el número de etiquetas necesario para la entidad en el campo *Number of Labels*, presionando luego sobre el botón *Refresh Arrivals* o *Refresh Labels*.

Los nombres de las etiquetas se pueden cambiar haciendo doble clic en el encabezado de las columnas agregadas después de *Quantity*. En la primera etiqueta *Entidad* se coloca el número de entrada correspondiente, las dos etiquetas restantes se dejan en cero. Por último se presiona *Apply* u *OK* para aplicar los cambios.

Para la creación de la tabla global del tiempo de servicio, se accede al menú *Tools*, opción *Global Tables* y posteriormente en *Add*, tal como se ve a continuación:

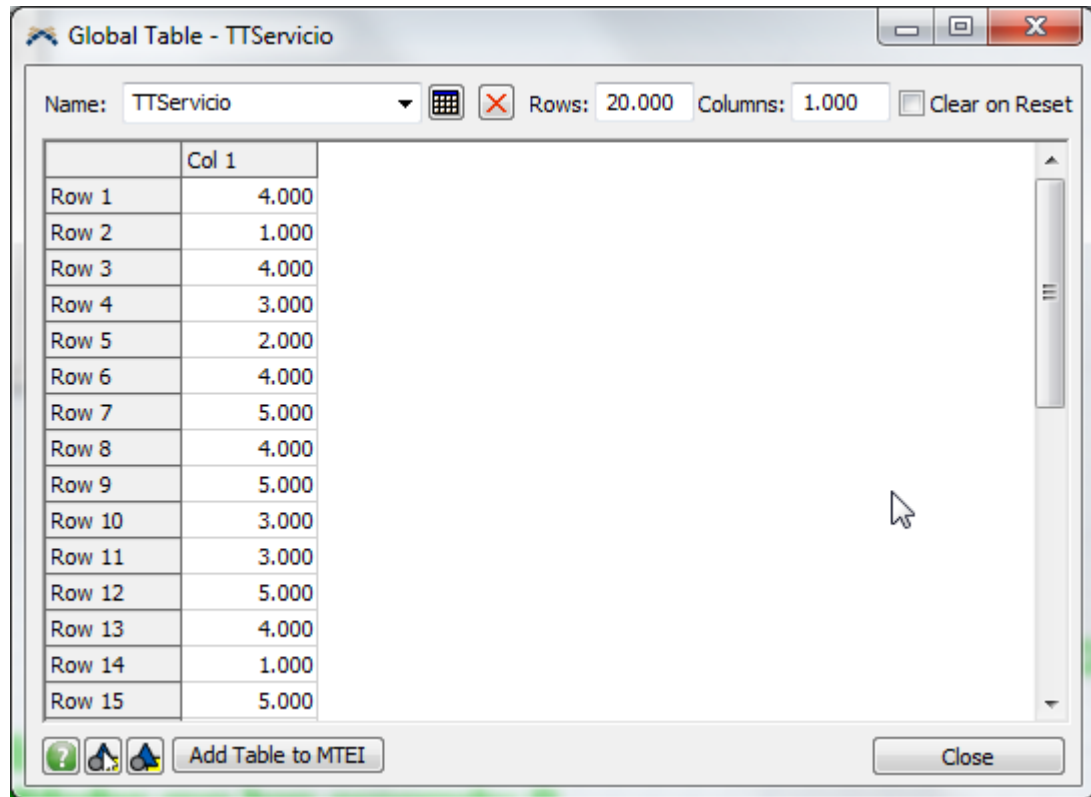
**Figura 8. Creación de tablas globales en el menú Tools**



Se ingresan el número de filas y de columnas de la tabla, además del nombre.

Los datos se encuentran en la columna Tiempo de servicio de la tabla 4.

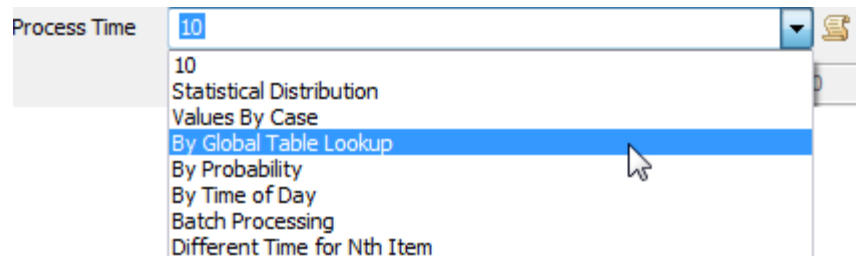
**Figura 9. Creación de tablas globales en el menú Tools**

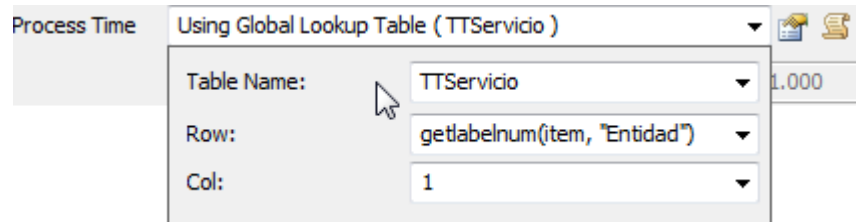


**6. Asignación del tiempo de proceso para SERVIDOR:**

Se configura el tiempo de proceso en el *Processor* o en el *SERVIDOR* accediendo a sus propiedades con doble clic y luego en la sección *Process Time* desplegando la lista de opciones.

**Figura 10. Tiempo de proceso por búsqueda en Tabla Global**



**Figura 11. Configuración del tiempo de proceso del SERVIDOR**

Process Time Using Global Lookup Table ( TTServicio )

Table Name: TTServicio 1.000

Row: getlabelnum(item, "Entidad")

Col: 1

Dado que la fila (*Row*) en la tabla hace referencia a la entidad que ingresa al objeto *Processor*, esta varía con cada entrada, por lo cual se necesita de una instrucción que registre el número de entidad entrante para que pueda buscar su tiempo de proceso en la primera columna de la tabla global.

Las instrucciones que se pueden emplear obteniendo el mismo resultado son:

- `getinput(current)`  
Retorna el número de entradas o de entidades que han ingresado a *current*, donde *current* hace referencia al objeto que está ejecutando esa instrucción. Se activa al ingresar una entidad al objeto fijo.
- `getlabelnum(item, "Entidad")`  
Regresa el valor numérico de una etiqueta perteneciente a un objeto de Flexsim. Su sintaxis es `getlabelnum(objeto, Etiqueta)` donde Etiqueta debe ir en comillas dobles: "*Entidad*".

Objeto hace referencia a un elemento de Flexsim, si son entidades por lo general se denotan como *item* y si se necesita hacer referencia al mismo objeto que ejecuta la instrucción es *current*, pero si es a otro objeto se utiliza una estructura de nodo que se verá en talleres posteriores.

También puede utilizarse la siguiente instrucción con una ligera modificación:

- `getitemtype(item)`

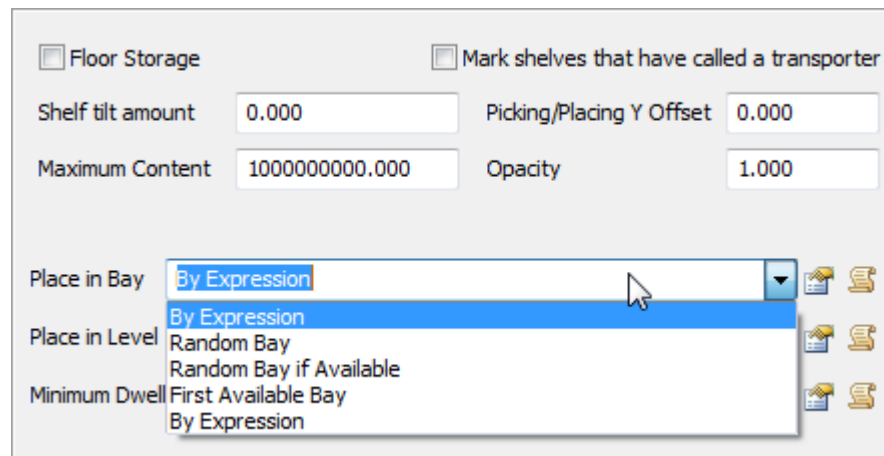
Obtiene el itemtype de una entidad o de un item, en este caso de los flowitems que ingresan al objeto fijo.

El itemtype es el principal atributo de una entidad, es una etiqueta también pero que siempre está presente en los flowitem. Si en la tabla de la figura 7 se ingresan los flowitems consecutivos del 1 al 20 tal como está en la etiqueta Entidad, se podría utilizar esta instrucción, sin embargo como en todos está el mismo valor 1, siempre regresaría 1 y buscaría en la tabla global el mismo tiempo de la entidad 1 que equivale a 4 minutos.

## 7. Salida de las entidades del sistema:

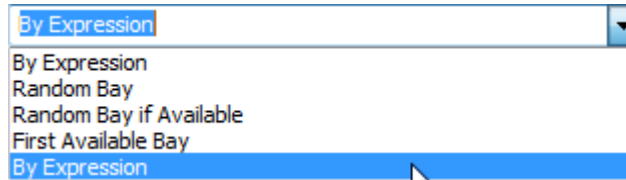
Se puede realizar con un *Sink* o con un *Rack* para conservar las entidades y poder ver los valores de sus etiquetas. En este ejercicio se almacenarán, por lo tanto se debe configurar las propiedades del *Rack* o ALMACEN – SALIDA:

**Figura 12. Configuración de propiedades en el Rack**

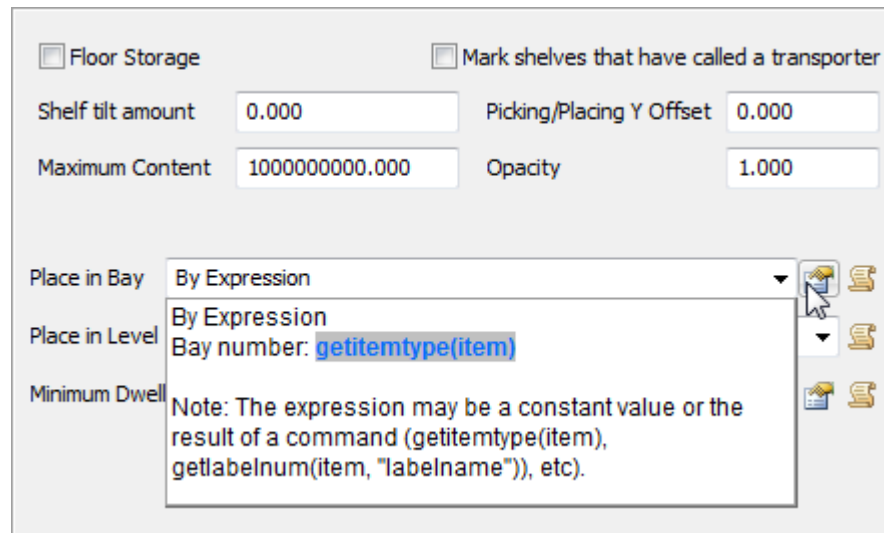


Tanto para la bahía (columnas) como para los niveles (filas) del rack, se debe escoger la misma opción *By Expression* y proceder de la siguiente forma:

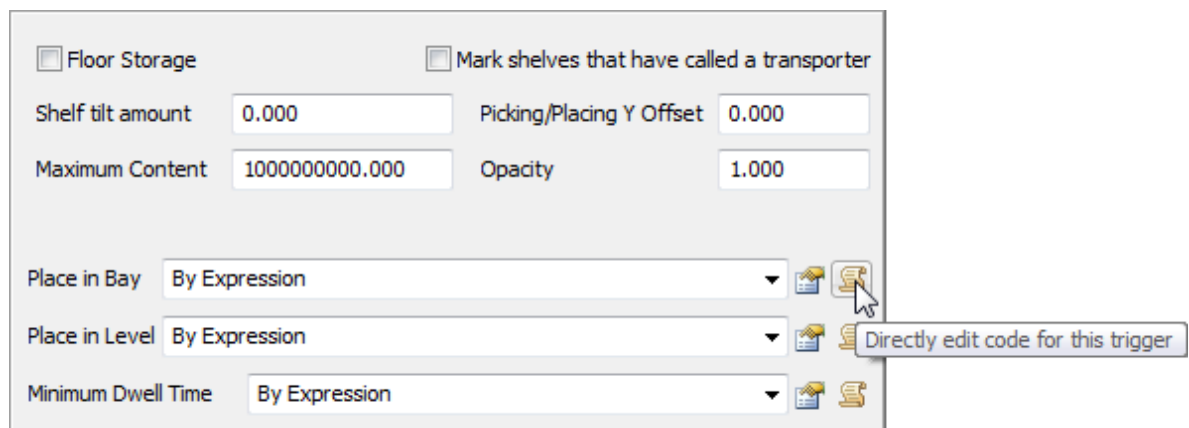
**Figura 13. Configuración de bahías y niveles del Rack**



**Figura 14. Edición de parámetros para la opción escogida**



**Figura 15. Edición de código para el controlador seleccionado**



Los anteriores pasos desde la figura 12 hasta la 15 deben ser iguales para las secciones *Place in Bay* y *Place in Level* del rack. Posteriormente aparecerá una ventana con el código de programación del *trigger* o controlador seleccionado como se aprecia en la figura 16 y se cambiará por el código de la figura 17:

**Figura 16. Código para la sección Place in Bay del Rack**

```

ALMACEN SALIDA - Place In Bay | ALMACEN SALIDA - Place In Level
1 treenode item = parnode(1);
2 treenode current = ownerobject(c);
3 /**By Expression*/
4 /** \nBay number: */
5 int baynum = /**/getitetype(item)/**/;
6 /** \n\nNote: The expression may be a constant value or the result of a command (getitetype(
7
8 return max(1, min(rackgetnrofbaies(current), baynum));
9
10 // note: the "contenttable" node contains "bay" nodes, and the "bay" nodes contain "level" no

```

**Figura 17. Edición del código en la sección Place in Bay**

```

ALMACEN SALIDA - Place In Bay | ALMACEN SALIDA - Place In Level
1 treenode item = parnode(1);
2 treenode current = ownerobject(c);
3 /**By Expression*/
4 /** \nBay number: */
5 int baynum = /**/getlabelnum(item, "Entidad")/**/;
6 if (baynum % 10 != 0)
7     baynum = baynum/10+ 1;
8 else
9     baynum /= 10;
10 /** \n\nNote: The expression may be a constant value or the result of a command (getitetype(
11
12 return max(1, min(rackgetnrofbaies(current), baynum));
13
14 // note: the "contenttable" node contains "bay" nodes, and the "bay" nodes contain "level" no

```

El anterior cambio obedece a que por la opción escogida las entidades se almacenan en las bahías según el itemtype, es decir, si una entidad tiene un flowitem 5 es almacenado en la bahía 5, pero como no se está utilizando el itemtype, es necesario modificar el código para trabajar con etiquetas y para que las entidades se almacenen de a diez por bahía. Se asume un rack de 10 bahías (columnas) por 10 niveles (filas).

La instrucción `getlabelnum(item, "Entidad")` captura el número de entidad al momento de ejecución del código, es decir, al entrar al *Rack*, esta instrucción capturaría números entre 1 y 20.

```
5 int baynum = /**/getlabelnum(item, "Entidad")/**/;  
6 if (baynum % 10 != 0)  
7     baynum = baynum/10+ 1;  
8 else  
9     baynum /= 10;
```

La sentencia si – sino (if – else) trabaja evaluando una condición lógica, si es verdadera se ejecutan las instrucciones después del if, sino se ejecutan las que se encuentran después del else.

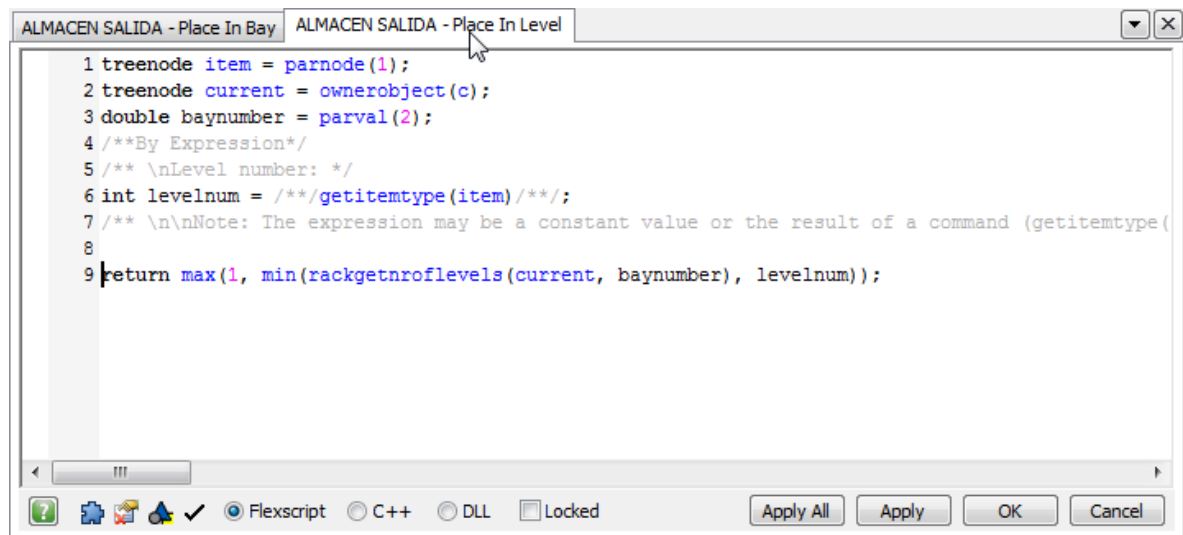
Una vez capturada el número de entidad en la variable entera baynum, se realiza la operación mod o módulo de la división o residuo. Ejemplo:  $34 \% 7 = 6$ , porque 34 dividido en 7 da 4 y sobran 6, este residuo es el resultado de la operación mod. Si se aplica la operación mod entre un número cualquiera y 10, da como resultado 0 si es múltiplo, ejemplo  $10 \% 10 = 0$ ;  $20 \% 10 = 0$ , de lo contrario da un valor entre 1 y 9, ejemplo  $5 \% 10 = 5$ ;  $17 \% 10 = 7$ .

A los múltiplos (mod da 0) se les aplica una división entera entre 10 (instrucción después del else), es decir, la entidad  $10/10 = 1$ , se almacena en la bahía 1, la entidad  $20/10 = 2$ , se almacena en la bahía 2.

Ahora si no es múltiplo (mod da diferente de cero  $\neq 0$ , instrucción en seguida del if) se divide entre 10 y se suma una unidad. Ejemplo: 5 no es múltiplo de 10, luego  $5\%10 = 5$ , por lo tanto se hace la división entera entre 10 y se le suma 1,  $5/10 + 1 = 0 + 1 = 1$ , es decir se almacena en la bahía 1. La entidad número 17 no es múltiplo de 10, luego  $17\%10 = 7$ , por lo tanto se realiza la operación  $17/10 + 1 = 1 + 1 = 2$ , luego se almacena en la bahía 2.

De la misma manera se procede con el *trigger Place in Level* del Rack:

Figura 18. Código para la sección Place in Level del Rack



```
ALMACEN SALIDA - Place In Bay  ALMACEN SALIDA - Place In Level
1 treenode item = parnode(1);
2 treenode current = ownerobject(c);
3 double baynumber = parval(2);
4 /**By Expression*/
5 /** \nLevel number: */
6 int levelnum = /**/getitemtype(item)/**/;
7 /** \n\nNote: The expression may be a constant value or the result of a command (getitemtype(
8
9 return max(1, min(rackgetnrolevels(current, baynumber), levelnum));
```

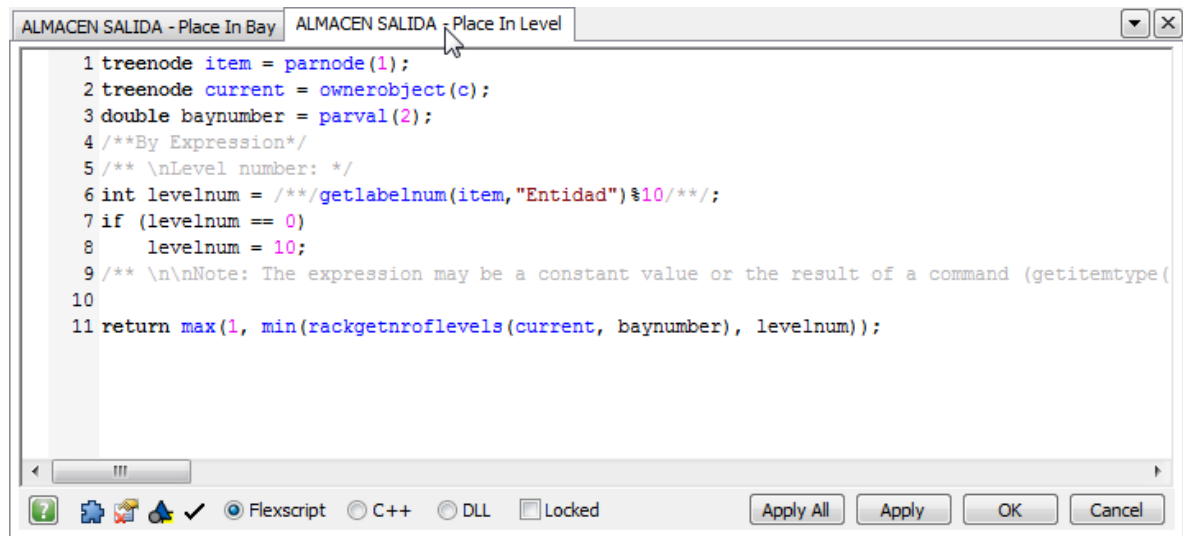
Deberá ser cambiado por el siguiente código, a partir de la línea 6:

```
6 int levelnum = /**/getlabelnum(item, "Entidad")%10/**/;
7 if (levelnum == 0)
8     levelnum = 10;
```

Se debe capturar la etiqueta Entidad mod 10 y no el ItemType, es decir, el residuo del valor Entidad dividido en 10. Ejemplo: entidad  $7\%10 = 7$ , entidad  $15\%10 = 5$ , que nos indica el nivel en el que debe ser almacena esa entidad.

Ahora si el valor es múltiplo de 10, es decir, la operación mod da cero, por lo tanto se debe almacenar en el décimo nivel.

**Figura 19. Edición del código en la sección Place in Level**



```
1 treenode item = parnode(1);
2 treenode current = ownerobject(c);
3 double baynumber = parval(2);
4 /**By Expression*/
5 /** \nLevel number: */
6 int levelnum = /**/getlabelnum(item, "Entidad")$10/**/;
7 if (levelnum == 0)
8     levelnum = 10;
9 /** \n\nNote: The expression may be a constant value or the result of a command (getitemtype(
10
11 return max(1, min(rackgetnrolevels(current, baynumber), levelnum));
```

## 8. Creación de Etiquetas:

Se definen unas etiquetas o *labels* en los 4 objetos del modelo, con el fin de hallar unas estadísticas de interés y familiarizarse con la creación y utilización de etiquetas.

No es necesario emplear esas etiquetas para las estadísticas que se quieren calcular manualmente, porque Flexsim arroja unas estadísticas básicas mientras se está simulando el modelo, es decir, en tiempo real de simulación, pero el fin es contrastar resultados de simulación en Excel vs. resultados en Flexsim que pueden ser determinados manualmente o mediante la recolección de estadísticas de los objetos en el módulo *Flexsim Chart*.

**Tabla 6. Etiquetas de los objetos del modelo**

Objeto	Etiqueta	Descripción
<b>Flowitems</b>	Entidad	Número de entidad
	TCola	Tiempo en Cola que permanece la entidad
	TCiclo	Tiempo de Ciclo para el flowitem
<b>Source</b>	Comienzo	Tiempo de llegada de la primera entidad (únicamente)
	Entidad	Número de entidad
	Llegada	Tiempo de llegada o de arribo
	InterArrivalTimeProm	Tiempo entre llegadas promedio
<b>Queue</b>	ClientesEsperan	Número de clientes que han esperado en cola
	ClientesCola	Entidades que se encuentran en un tiempo dado en cola
	SumTEspera	Sumatoria del tiempo de espera
	TEsperaPromedio	Tiempo de espera promedio
	TMEsperaE	Tiempo medio de espera de los que esperan
	ProbEspera	Probabilidad de espera
<b>Processor</b>	Entidad	Número de entidad en el servidor
	SumTO	Sumatoria del tiempo de operación
	FU	Factor de utilización del servidor
	Idle	Porcentaje de inactividad del servidor
<b>Rack</b>	TFS	Tiempo de finalización de la simulación

El factor de utilización y el tiempo de inactividad de un objeto son calculados de manera automática por Flexsim, no necesitan calcularse manualmente, pero se hace el ejercicio para adquirir destreza en el uso de código de Flexsim y utilización de variables locales, globales y etiquetas de objetos.

Lo mismo ocurre con el tiempo de espera promedio y el número de entidades que han ingresado a un objeto.

## 9. Configuración de Triggers:

Los *Triggers* son controladores o disparadores de eventos de los objetos en Flexsim. Se encuentran en la pestaña Triggers de las propiedades del objeto y algunos en la pestaña general o principal de objetos fijos en Flexsim.

Un ejemplo de *trigger* común a todos los objetos es *OnReset* que se ejecuta cuando se resetea o reinicia el modelo. *OnEntry* cuando entra una entidad al objeto, lo mismo cuando sale la entidad con el trigger *OnExit*, que son los controladores comunes a todos los objetos. Hay algunos específicos para cada objeto. Para el ejercicio solamente serán necesarios los disparadores *OnEntry* y *OnExit*.

Todos los controladores o *triggers* tienen código de programación predeterminado, dependiendo de la acción a ejecutar en el disparador, lo que hace más fácil la edición de código.

Como norma se acostumbra a comentar líneas de código de la siguiente manera:

```
inc(label(current,"ClientesCola"),1); //Incrementar en 1 la etiqueta ClientesCola
```

El comentario esta después del doble // en color verde, quiere decir que Flexsim no compilará el texto a la derecha. Otra forma de comentario es el que está comprendido entre los siguientes símbolos:

```
/*  
Esto es un comentario  
*/
```

En el objeto COLA es de interés conocer el número de clientes que han esperado en cola en determinado momento, el tiempo en cola de las entidades del sistema, tener estadísticas del tiempo medio de espera y calcular la probabilidad de que una entidad tenga que esperar.

Por lo tanto es necesario capturar el tiempo de entrada y de salida de la cola y hallar su diferencia; si es mayor a cero quiere decir, que la entidad esperó en la cola, sino pasó derecho al servidor sin tener que esperar en la cola. El tiempo actual de simulación se captura con la función `time()` y la asignación de un valor a una etiqueta se hace con la instrucción:

`setlabelnum(objeto, etiqueta, valor)`

Asigna un valor numérico a una etiqueta del objeto pasado como primer parámetro. En este caso se desea guardar el valor del tiempo de entrada a la cola en la etiqueta *TCola* del objeto *item*, que hace referencia a los *flowitems* que pasan por la cola. Posteriormente se incrementa la etiqueta *ClientesCola* del objeto COLA.

Figura 20. Trigger OnEntry del objeto COLA

```
1 treenode item = parnode(1);
2 treenode current = ownerobject(c);
3 int port = parval(2);
4
5 { ***** PickOption Start *****\
6 /**popup:SetCombo*/
7 /**tag:title/**Set Label Value*/
8 /**tag:command/**/setlabelnum/**/ {
9 /**tag:item/**/item/**/,
10 /**tag:var/**/"TCola"/**/
11 /**tag:value/**/,time (**/
12 );
13 /**\n\n*/
14 inc(label(current,"ClientesCola"),1);
15 } ***** PickOption End *****\
16
```

Figura 21. Trigger OnExit del objeto COLA

```
1 treenode item = parnode(1);
2 treenode current = ownerobject(c);
3 int port = parval(2);
4
5 { //***** PickOption Start *****\\
6 /**popup:SetCombo*/
7 /**tag:title/**Set Label Value*/
8 /**tag:command/**/setlabelnum/**/(
9 /**tag:item/**/item/**/,
10 /**tag:var/**/"TCola"/**/
11 /**tag:value/**/,time()-getlabelnum(item,"TCola")/**/
12 );
13 /**\n\n*/
14 if (getlabelnum(item,"TCola")>0)
15 {
16     inc(label(current,"ClientesEsperan"),1);
17     inc(label(current,"SumTEspera"),getlabelnum(item,"TCola"));
18     setlabelnum(current,"TMEsperaE",getlabelnum(current,"SumTEspera")/getlabelnum(current,"ClientesCola"));
19 }
20 setlabelnum(current,"ProbEspera",getlabelnum(current,"ClientesEsperan")/getlabelnum(item,"Entidad"));
21 setlabelnum(current,"TEsperaPromedio",getlabelnum(current,"SumTEspera")/getlabelnum(item,"Entidad"));
22 inc(label(current,"ClientesCola"),-1);
23 //setlabelnum(current,"TEsperaPromedio",get(stats_staytimeavg(current)));
24 } //***** PickOption End *****\\
```

En el trigger *OnExit* nuevamente se captura el tiempo actual de simulación al salir la entidad de COLA pero restándole el valor de tiempo guardado a la entrada del objeto, es decir, se está realizando una diferencia del tiempo de salida menos el tiempo de entrada para saber la duración en el objeto. Posteriormente como este disparador de evento es de salida, se resta una unidad a su etiqueta de *ClientesCola* al final del controlador.

Las otras operaciones realizadas en este trigger (ver figura 21) dentro de la condición if y después de esta, hacen referencia a la etiqueta *ClientesEsperan* que se incrementa en uno si el tiempo de espera en cola es mayor que cero, en este caso si el valor de *TCola* > 0.

La etiqueta *SumTEspera* y *TMEsperaPromedio* se hallan con un promedio de los tiempos de espera de las entidades, por lo tanto, se necesita totalizar o acumular los tiempos de espera y dividirlos por el número de entidades que han pasado hasta ese instante para determinar el tiempo de espera promedio.

El otro trigger empleado en la simulación es *OnProcessFinish* del objeto SERVIDOR que es un processor. Las operaciones realizadas son los cálculos para hallar el factor de utilización del servidor manualmente. Estas operaciones incluyen acumular en una etiqueta los tiempos de servicio de todas las entidades que pasan por el servidor y dividir es acumulativo en el tiempo transcurrido. En la figura 22 se observan esas operaciones en el código del trigger *OnProcessFinish* del objeto *SERVIDOR*.

**Figura 22. Trigger OnProcessFinish del objeto SERVIDOR**

```
1 /**Custom Code*/
2 treenode item = parnode(1);
3 treenode current = ownerobject(c);
4
5 int Entidad; double SumaTO;
6
7 Entidad = getlabelnum(item,"Entidad");
8 setlabelnum(current,"Entidad",Entidad);
9
10 inc(label(current,"SumTO"),gettablenum("TTServicio",Entidad,1));
11 SumaTO = getlabelnum(current,"SumTO");
12
13 setlabelnum(current,"FU", (SumaTO/time())*100);
14 setlabelnum(current,"Idle",100-getlabelnum(current,"FU")); stop();
```

Se agrega al final la instrucción `stop()` para detener la simulación y observar con detenimiento como evoluciona la simulación.

Por último queda por configurar el *Trigger OnEntry* del objeto *ALMACEN SALIDA* para capturar el tiempo de ciclo y hallar el tiempo promedio del sistema.

Figura 23. Trigger OnExit del objeto ALMACEN SALIDA

```
1 treenode item = parnode(1);
2 treenode current = ownerobject(c);
3 int port = parval(2);
4
5 { //***** PickOption Start *****\\
6 /**popup:SetCombo*/
7 /**tag:title**/**Set Label Value*/
8 /**tag:command**/**setlabelnum/**/ (
9 /**tag:item**/**item/**/,
10 /**tag:var**/**"TCiclo"/**/
11 /**tag:value**/**, time()-getlabelnum(item,"TCiclo")/**/
12 );
13 /**\n\n*/
14 TC = getlabelnum(item,"TCiclo");
15 SumTC += TC;
16 TMF = SumTC / content(current);
17
18 TFS = time();
19 setlabelnum(current,"TFS",time());
20
21 if (getlabelnum(item,"Entidad")==gettablerows("TEntreLlegadas"))
22     stop();
23 } //***** PickOption End *****\\
```

El código guarda en la etiqueta TCiclo de la entidad, el tiempo de ciclo como la operación de capturar el tiempo en que entra al Rack menos el tiempo que tenía almacenado, que es el de entrada al sistema, operación realizada con la instrucción: `setlabelnum(item,"TCiclo",time-time()-getlabelnum(item,"TCiclo"))`;

Las líneas 14 a 16 calculan el tiempo de flujo medio o tiempo promedio en el sistema, totalizando los tiempos de ciclo `SumTC += TC` y dividiendo por el número de entidades que salen del sistema o que entran al Rack: `content(current)`

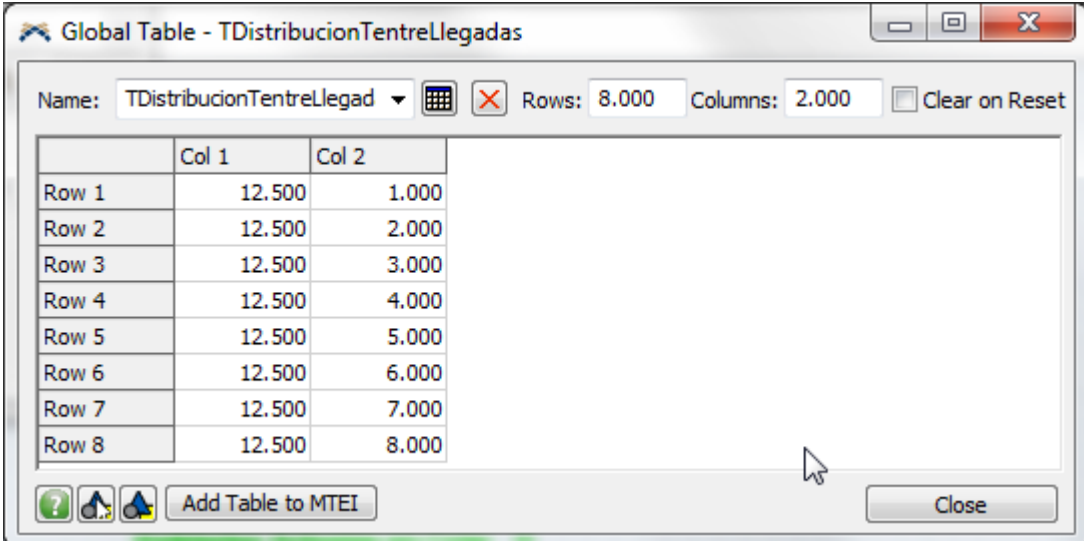
Las líneas 21 a 23 no son necesarias, lo que hacen es evaluar si la entidad que entra al almacén es la 20 para detener la simulación con la instrucción `stop()` y de esta manera comparar los resultados de Flexsim y de Excel.

Una vez realizados los pasos anteriores se corre la simulación, observando que al salir una entidad del servidor y dirigirse al almacén se detiene la simulación, lográndose leer las estadísticas de interés del modelo en los *VisualTool* y de esta manera se puede comparar con las arrojadas por el archivo Excel adjunto al taller.

Este modelo simula 20 entidades, sin embargo, este número es muy pequeño para sacar conclusiones fiables. La exactitud de los resultados de la simulación mejorará en la medida en que se aumenta el tamaño muestral o se corre la simulación por un tiempo suficiente hasta que alcance el estado estable (estacionario) en sistemas no terminales o durante todo el período de simulación de un sistema terminal.

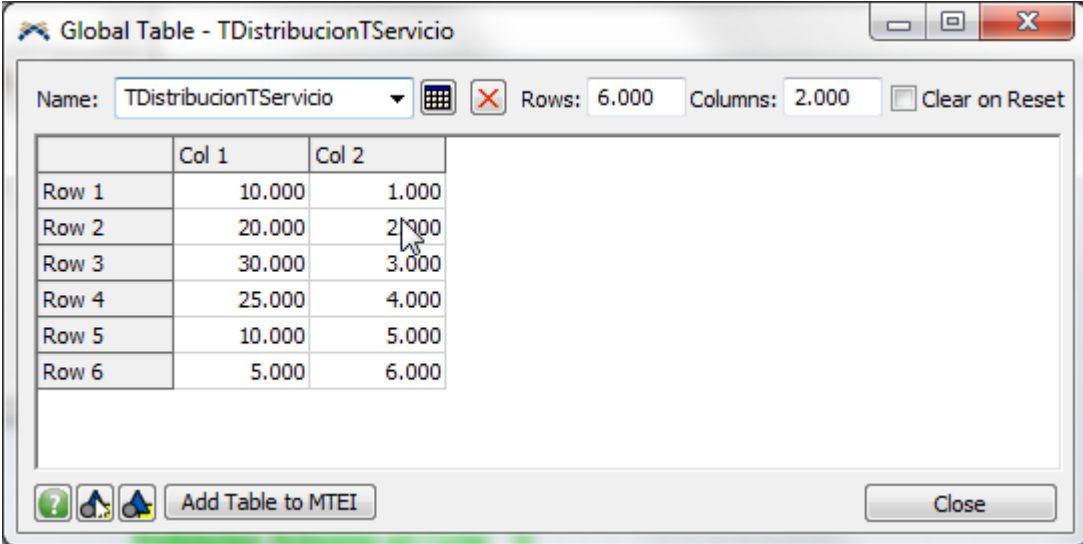
Para finalizar la simulación del ejercicio guiado, se debe cambiar el tiempo entre arribos y el tiempo de servicio por las distribuciones empíricas que los caracterizan en vez de los tiempos de las 20 observaciones realizadas en Excel, por lo tanto se deben crear dos tablas que representen las distribuciones empíricas como se ilustra en las dos siguientes figuras:

**Figura 24. Tabla global de la distribución del tiempo entre llegadas**



	Col 1	Col 2
Row 1	12.500	1.000
Row 2	12.500	2.000
Row 3	12.500	3.000
Row 4	12.500	4.000
Row 5	12.500	5.000
Row 6	12.500	6.000
Row 7	12.500	7.000
Row 8	12.500	8.000

Figura 25. Tabla global de la distribución del tiempo de servicio



	Col 1	Col 2
Row 1	10.000	1.000
Row 2	20.000	2.000
Row 3	30.000	3.000
Row 4	25.000	4.000
Row 5	10.000	5.000
Row 6	5.000	6.000

Una distribución empírica es una tabla que contiene una lista de valores y porcentajes de probabilidad asociados con cada valor. En la primera columna de la tabla deben ir los porcentajes como números entre 0 y 100 y que totalicen 100, y en la segunda columna sus valores asociados.

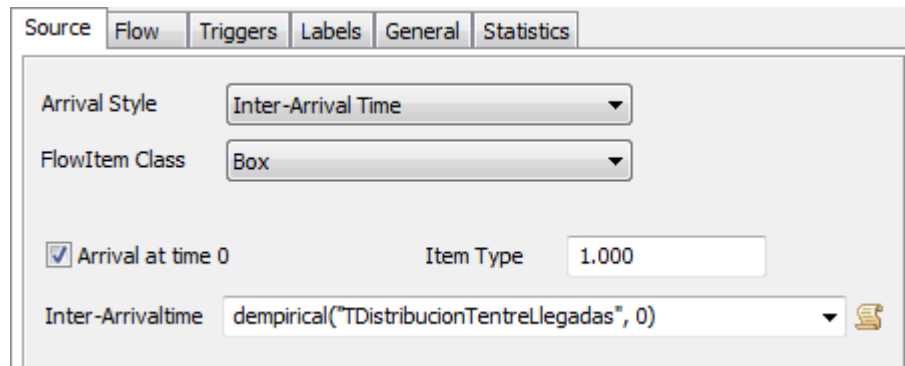
Flexsim tiene 3 comandos para distribuciones empíricas:

1. *dempirical* para valores discretos.
2. *cempirical* para valores continuos empezando desde 0.
3. *empirical* para valores continuos empezando desde el primer valor de probabilidad.

Según los datos de la tabla de la figura 24 todos los valores tienen la misma probabilidad de ocurrencia del 12.5%. En los datos de la figura 25 el 30% tienen el valor de tiempo de servicio de 3 minutos y el 5% 6 minutos.

Posteriormente en la fuente de llegadas se cambia el tipo de llegada de *Arrival Schedule* a *Inter-Arrival Time* y haciendo referencia a la tabla creada en la figura 24 (*TDistribucionTentreLlegadas*) como se observa en la figura 26:

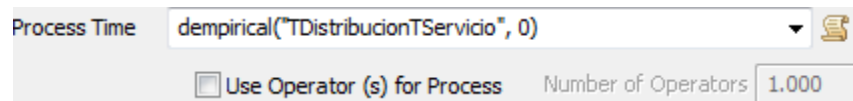
**Figura 26. Distribución empírica en las llegadas**



The screenshot shows the 'Source' configuration window in FlexSim. The 'Arrival Style' dropdown is set to 'Inter-Arrival Time'. The 'FlowItem Class' dropdown is set to 'Box'. The 'Arrival at time 0' checkbox is checked. The 'Item Type' text box contains '1.000'. The 'Inter-Arrivaltime' dropdown is set to 'dempirical(\"TDistribucionTentreLlegadas\", 0)'. The window has tabs for 'Source', 'Flow', 'Triggers', 'Labels', 'General', and 'Statistics'.

Se procede de igual forma con el servicio de las entidades:

**Figura 27. Distribución empírica en el servicio**



The screenshot shows the 'Process Time' configuration window in FlexSim. The 'Process Time' dropdown is set to 'dempirical(\"TDistribucionTServicio\", 0)'. The 'Use Operator (s) for Process' checkbox is unchecked. The 'Number of Operators' text box contains '1.000'.

Código correspondiente a la figura 27:

```
1 treenode current = ownerobject(c);  
2 treenode item = parnode(1);  
3 return /**/dempirical("TDistribucionTServicio", 0)**direct**/;
```

Donde, `dempirical("TDistribucionTServicio", 0)` es la instrucción para utilizar una distribución empírica discreta cuyos valores y probabilidades se encuentran en una tabla global. El parámetro 0 hace referencia a la semilla de aleatorios empleada.

## 10. Configuración de los VisualTool:

Los *VisualTool* son objetos que muestran texto, imágenes o un gráfico en pantalla, estos objetos sirven de apoyo al modelador para visualizar información en tiempo real de simulación, tal como las variables globales o una etiqueta de alguna entidad. En este ejemplo se emplean 3 variables globales para calcular el tiempo de ciclo de una entidad, tal como se observa en las operaciones de la figura 23, sin embargo, el contenido de estas variables no se visualiza en pantalla y ahí es donde surge una de las utilidades de este objeto.

Como se dese mostrar el tiempo medio de flujo calculado anteriormente, se agrega un *VisualTool* al layout y en las propiedades de este objeto en la lista desplegable correspondiente a la opción *Visual*, se selecciona *Text*, apareciendo un área correspondiente al texto a introducir, para lo cual, se presiona en el botón de edición de código para editarlo y agregar las siguientes líneas:

```
1 treenode current = ownerobject(c);
2 treenode textnode = parnode(1);
3 /**Display Object Stat*/
4 /** \nText: */
5 string starttext = /**/"Tiempo Medio de Flujo: "*/;
6 /** \nVariable: */
7 string statstr = numtostring(**/TMF/**/,0,2);
8
9 setnodestr(textnode, concat(starttext, statstr));
10
```

El texto a mostrar se compone de una leyenda o texto descriptivo seguido de dos puntos y un valor numérico correspondiente a la variable *TMF* que contiene el tiempo medio de flujo. El texto descriptivo se almacena como un tipo de dato *string* o de cadena de caracteres en la variable local *starttext*, el valor numérico se convierte a *string* con la función *numtostring* y se almacena en la variable *statstr*, posteriormente se concatenan los dos textos con la función *concat*.

## 2. TALLER A DESARROLLAR

### OBJETOS DEL MODELO

Tabla 7. Objetos del Taller No. 2

Elemento	Objeto	Nombre	Puertos	
			Entrada	Salida
Llegadas	Source	FUENTE	0	1
Operaciones	Processor	ESTACION 1	1	1
		ESTACION 2	1	1
Filas	Queue	ENTRADA	1	1
		SALIDA	1	0

Figura 28. Esquema inicial del Taller No. 2



### DURACIÓN DEL TALLER

La duración estimada del taller es de 4 horas.

## DESCRIPCIÓN DEL TALLER

Simular en FLEXSIM el siguiente ejercicio adaptado de Chase et al. (2009) de una línea de ensamble de dos etapas (línea de espera en serie) y contrastar los resultados teóricos (Excel) y los simulados:

Se considera una línea de ensamble que fabrica un producto de tamaño físico significativo, como un refrigerador, estufa, automóvil, lancha, televisión o muebles, a través de dos estaciones de trabajo en serie.

El tamaño del producto es una consideración importante del análisis de la línea de ensamble y el diseño porque la cantidad de productos en cada estación de trabajo afecta el desempeño del trabajador. Si el producto es grande, entonces las estaciones de trabajo dependen entre sí. Por ejemplo, la producción de la estación 1 alimenta a la estación 2. Si las estaciones están contiguas de modo que no hay espacio en medio para las piezas, entonces la estación 1, de trabajar despacio, haría esperar a la estación 2. Si, por el contrario, la estación 1 termina rápido (o si la estación 2 tarda más en terminar la tarea), entonces debe esperar a la estación 2. En esta simulación, se supone que la estación 1, puede sacar una pieza nueva para trabajarla cuando sea necesario.

Hallar:

- a) El porcentaje de utilización de la estación 1 y de la estación 2.
- b) El tiempo promedio de espera.
- c) El tiempo promedio de desempeño o de servicio.
- d) Tiempo promedio por unidad.
- e) Tiempo medio de flujo o tiempo promedio de ciclo.

A continuación se encuentran los intervalos de números aleatorios para los tiempos de servicio de la estación 1 y de la estación 2.

**Tabla 8. Intervalos de números aleatorios de la Estación 1**

Segundos	Frecuencias de Tiempos de Operación	Intervalo de Números Aleatorios
10	4	00 – 03
20	6	04 – 09
30	10	10 – 19
40	20	20 – 39
50	40	40 – 79
60	11	80 – 90
70	5	91 – 95
80	4	96 – 99
<b>Observaciones</b>	<b>100</b>	

**Tabla 9. Intervalos de números aleatorios de la Estación 2**

Segundos	Frecuencias de Tiempos de Operación	Intervalo de Números Aleatorios
10	4	00 – 07
20	5	08 – 17
30	6	18 – 29
40	7	30 – 43
50	10	44 – 63
60	8	64 – 79
70	6	80 – 91
80	4	92 – 99
<b>Observaciones</b>	<b>50</b>	

Para la simulación en hoja de cálculo se emplean los siguientes aleatorios:

Tabla 10. Números aleatorios para el ejercicio

Aleatorio T. Operación Estación 1	Aleatorio T. Operación Estación 2
93	1
52	45
15	72
64	35
86	2
21	82
83	31
89	13
69	53
41	48
32	13
1	67
11	91
2	76
11	41
55	34
18	28
39	53
13	41
7	21
29	54
58	39
95	70
27	60
59	93
85	51
12	35
34	51
60	87
97	29

Ejemplo de la tabla de simulación con 10 entidades o piezas:

**Tabla 11. Resultados de simulación en Excel**

Número Pieza	Número Aleatorio Estación 1	Tiempo Inicio Estación 1	Tiempo de Operación Estación 1	Tiempo de Finalización Estación 1	Tiempo de Espera Estación 1	Número Aleatorio Estación 2	Tiempo Inicio Estación 2	Tiempo de Operación Estación 2	Tiempo de Finalización Estación 2	Tiempo de Espera Estación 2
1	56	00	50	50		83	50	70	120	50
2	55	50	50	100	20	47	120	50	170	
3	84	120	60	180		08	180	20	200	10
4	36	180	40	220		05	220	10	230	20
5	26	220	40	260		42	260	40	300	30
6	95	260	70	330		95	330	80	410	30
7	66	330	50	380	30	17	410	20	430	
8	3	410	10	420	10	21	430	30	460	
9	57	430	50	480		31	480	40	520	20
10	69	480	50	530		90	530	70	600	10
<b>Suma</b>			<b>470</b>		<b>60</b>			<b>430</b>		<b>170</b>

## TALLER 3: ANÁLISIS DE DATOS DE ENTRADA CON EXPERTFIT

### INTRODUCCIÓN

La recolección y análisis de datos en un estudio de simulación es una de las etapas preliminares y más laboriosas, donde se recopila información estadística de las variables aleatorias del modelo. En esta etapa se debe establecer qué información es útil para la determinación de las distribuciones de probabilidad asociadas a cada una de las variables aleatorias necesarias para la simulación. Al finalizar la recolección y análisis de datos para todas las variables del modelo, se tendrán las condiciones necesarias para generar una versión preliminar del problema que se está simulando.

### OBJETIVOS

- ❖ Determinar la distribución de probabilidad a partir de un conjunto de datos utilizando la herramienta o módulo ExpertFit del software Flexsim.
- ❖ Seleccionar una distribución de probabilidad en ausencia de datos.

## MARCO TEÓRICO

### Recolección y análisis de datos

En todo proyecto de simulación, después de formular el problema y definir el sistema en términos de un modelo conceptual (o mental de cómo se configura el sistema y cómo opera), se genera un modelo de simulación base, no demasiado detallado, pues en esta instancia no se dispone de mucha información y de manera paralela se comienza con la etapa de recolección y análisis de datos. En esta etapa se identifican, obtienen y analizan los datos necesarios para el estudio de simulación definiendo el sistema que se va a modelar. Este paso genera un documento de los datos de entrada (inputs) empleados en la simulación.

La tarea de recolectar datos para sistemas nuevos puede requerir mucho tiempo ya que los datos pueden no estar previamente definidos y se requerirá hacer estimaciones. Para sistemas existentes previamente, puede haber archivos de datos históricos pero aun así se puede requerir filtrar y organizar la información, además de saber si la información existente está actualizada para el modelo.

Los datos deben ser recolectados de manera organizada y localizando los datos que son de utilidad en la simulación.

Algunas recomendaciones para identificar la información importante y minimizar el esfuerzo de recolección son:

- **Identificar eventos disparadores:** cuando se definen las actividades del sistema, se deben identificar las causas o condiciones que las provocan. Se deben considerar preguntas como: ¿qué motiva el movimiento de entidades?, ¿qué conduce al uso de un recurso?

- **Análisis de sensibilidad:** Enfocarse en factores de impacto del sistema donde se deben obtener buenos datos, de calidad y eliminar aquellos que no aporten elementos importantes al desarrollo de la simulación, ya que se puede consumir tiempo importante, por ejemplo valores de tiempo de mantenimiento preventivo que no retrasan un proceso, pueden ignorarse si no son importantes para el propósito de la simulación.
- **Aislar tiempos de actividades:** sólo se debe tomar en cuenta el tiempo que toma la actividad misma, excluyendo tiempos adicionales a la actividad.
- **Buscar agrupaciones de datos comunes:** evitar el manejo de muchos tipos de datos diferentes que no se podrán manejar con el mismo detalle en la simulación.
- **Separar variables de entrada de variables de respuesta:** las variables de entrada definen cómo trabaja el sistema, las variables de respuesta indican cómo responde el sistema a un conjunto de variables de entrada.

### Pasos para la toma de datos

1. Determinar los requerimientos de datos.
2. Identificar las fuentes de datos.
3. Recolectar los datos.
4. Elaborar suposiciones donde sea necesario ante la ausencia de datos disponibles.
5. Análisis estadístico de los datos. Ajustar los datos obtenidos a una distribución teórica conocida mediante pruebas de bondad de ajuste.
6. Documentar y aprobar los datos.

## 1. Determinar los requerimientos de datos

El primer paso es determinar qué datos se requieren. Los datos se pueden categorizar en:

- *Datos estructurales*: representan todos los objetos del sistema modelado.
- *Datos operacionales*: explican cómo opera el sistema, cuándo, dónde y cómo toman lugar las actividades.
- *Datos numéricos*: proporcionan información cuantitativa sobre el sistema como capacidades, tiempos entre llegadas, tiempo entre fallas.

El uso de cuestionarios puede ser útil para recolectar de manera ordenada los diferentes tipos de datos. Ahorran tiempo en la recolección y pueden dar una mejor visión de la naturaleza de los datos al acudir al lugar y con las personas relacionadas con la generación de los mismos.

## 2. Identificar fuentes de información de datos

Por lo general, las fuentes de información de los datos que se requieren en la simulación son muy diversas. Es probable que se tengan que revisar reportes, hacer observaciones personales y hacer muchas suposiciones. Los tipos de fuentes de información variarán dependiendo de si se está trabajando con un sistema existente o uno que será creado basándose en los resultados de la simulación. Cuando se dispone de datos existentes es importante verificar si la información está actualizada.

La información se puede obtener de fuentes como las siguientes:

- Registros históricos.
- Documentación de sistemas.
- Observación personal.

- Entrevistas a personas que trabajen en el sistema.
- Comparación con sistemas similares.
- Estimados de diseño.
- Programas de operadores planeados.
- Información de los operarios del sistema.
- Investigación en la literatura relacionada.
- Diagramas de operaciones.
- Estudios de tiempos.
- Diagramas de flujo.
- Pronósticos de mercadeo.

### 3. Recolectar los datos

Al recolectar los datos se recomienda ir de lo general a lo específico. La recolección de datos continúa hasta el final del proceso de simulación conforme cambian los objetivos y la información que no estaba disponible empieza a materializarse.

La recolección debería realizarse en la siguiente secuencia:

- **Definir el flujo total de las entidades:** se da seguimiento al movimiento de las entidades dentro del sistema, se pueden usar diagramas de flujo de procesos y diagramas de recorrido.
- **Desarrollar una descripción de la operación:** define cómo las entidades se procesan a través del sistema, puede ser una narración descriptiva, una lista paso por paso o representarse en forma tabular.

- **Definir detalles adicionales:** Una vez que el modelo básico ha sido definido, se definen los detalles adicionales del proceso tales como el tiempo de caída de las máquinas (downtimes), tiempos de instalación y prioridades de trabajo. Esta información no es esencial en el modelo, pero es necesaria para elaborar un modelo que se asemeje a la realidad.

#### 4. Elaborar suposiciones

La simulación puede trabajar con datos incorrectos pero no puede funcionar con datos incompletos. En ocasiones es difícil poder obtener toda la información dado que la simulación frecuentemente trata de simular anticipando situaciones, es completamente válido y necesario hacer suposiciones sobre algunos datos, siempre y cuando se conozcan las implicaciones de las suposiciones y se llegue a una correcta interpretación de los resultados. Es aconsejable utilizar el menor número de ellas, dado que cuanto menos suposiciones tenga el modelo, más cercano a la realidad será.

#### 5. Análisis estadístico de datos

Para que los datos sean útiles a la simulación deben ser analizados e interpretados de tal forma que representen correctamente la operación del sistema. Las técnicas usadas para este fin son pruebas estadísticas que se ejecutan sobre los conjuntos de datos. Las pruebas más usadas son:

- Pruebas de independencia: se usan las gráficas de correlación y los diagramas de dispersión.
- Pruebas para datos distribuidos de manera idéntica o pruebas de uniformidad.

En este punto se ajustan los datos obtenidos a una distribución de probabilidad teórica conocida mediante pruebas de bondad de ajuste. Este ajuste de datos a una distribución de probabilidad se puede determinar mediante las pruebas *Chi-cuadrado* ( $\chi^2$ ), *Kolmogorov-Smirnov* (K-S) y *Anderson-Darling* (A-D), tests que son aplicados por las herramientas de análisis de datos del software de simulación.

Aunque el software de ajuste de datos (*ExpertFit* en Flexsim, *StatFit* en ProModel, *Input Analyzer* en Arena) realiza automáticamente el ajuste apropiado, lo primero que se debe hacer es el histograma de frecuencias (datos continuos) o el diagrama de barras (datos discretos), lo cual dará una idea de la distribución de probabilidad más cercana o apropiada, antes de realizar las pruebas respectivas.

Si el ajuste a una distribución teórica conocida no es apropiado o no supera las pruebas, se procede a utilizar distribuciones empíricas, donde los datos presentan características inusuales. Estas por lo general son multimodales. Se recomienda rechazar los valores de la muestra claramente atípicos o datos muy extremos, siempre y cuando la pérdida de información pueda ser asumida en el contexto del proceso que se está modelando.

Si el histograma tiene varias zonas dominantes se puede intentar separar y ajustar a una distribución diferente cada una de las zonas.

En la tabla siguiente se resume la aceptación en los test mencionados anteriormente:

**Tabla 12. Pruebas utilizadas para determinar el tipo de distribución de un conjunto de datos**

TEST	ESTADÍSTICO	ACEPTACIÓN	RECHAZO
Chi-cuadrado	$\chi^2 = \sum_{i=1}^r \frac{(O_i - E_i)^2}{E_i}$	$\chi^2 < \chi^2_{\alpha, r-p-1}$	$\chi^2 \geq \chi^2_{\alpha, r-p-1}$
Kolmogorov-Smirnov	$D = \text{máx}  POA_i - PEA_i $	$D \leq D_{\alpha, n}$	$D > D_{\alpha, n}$
Anderson-Darling	$A_n^2$	$A_n^2 < a_{\alpha, n}$	$A_n^2 \geq a_{\alpha, n}$

- Ho                    Distribución de probabilidad que se ajuste a la forma del histograma.  
r:                     Número de intervalos empleados.  
O<sub>i</sub>                    Frecuencia observada en el intervalo i.  
E<sub>i</sub>                    Frecuencia esperada en el intervalo i.  
(r-p-1)                Número de grados de libertad.  
p                      Número de parámetros de la distribución teórica que hay que estimar.  
POA<sub>i</sub>                Probabilidad observada acumulada en el intervalo i.  
PEA<sub>i</sub>                Probabilidad esperada acumulada en el intervalo i.

### Distribuciones de probabilidad para representar variables aleatorias

Se dividen en dos tipos:

- Teóricas: generan muestras con base en la formulación matemática. Ejemplo: la distribución exponencial, gamma.
- Empíricas: dividen los datos reales en grupos y calculan la proporción de valores en cada grupo.

Cada tipo de distribución se subdivide en:

- Continua: regresan cualquier cantidad de valor real (dentro de un rango para los tipos con límites). Por lo general se usan para representar duraciones de tiempo en un modelo de simulación.

- Discreta: regresan cantidades de valores enteros. A menudo se usa para describir el número de eventos que ocurren en un intervalo de tiempo.

Algunas de las funciones de probabilidad conocidas más utilizadas para generar los números aleatorios basados en su comportamiento son:

- Exponencial: para generar tiempos (de fila, de servicio).
- Normal: para generar valores como el consumo promedio de un cliente.
- Uniforme discreta: para representar el número resultante del lanzamiento de un dado, el número de cajero que atenderá el cliente).

Para algunas otras situaciones, ya se da establecido el comportamiento probabilístico y en otras habrá que hacer pruebas estadísticas para determinar la función de probabilidad que mejor representa dicho comportamiento, sin embargo, si no es posible ajustar los datos a una distribución teórica se emplea una *distribución empírica* que represente dichos datos continuos. Es recomendable tener un gran número de observaciones (mínimo 100) para elegir una correcta distribución.

### ¿Qué hacer si no hay datos disponibles?

En algunas ocasiones la información no podrá obtenerse debido a que: o el sistema no existe, o no puede obtenerse la información porque la recopilación es muy costosa, o que quizá no consigue la autorización que necesita. En estos casos suele emplearse una distribución triangular, si se conocen los valores extremos y el valor más probable o la moda, porque puede capturar los procesos con grados de variabilidad pequeños o grandes. Si los tiempos varían de forma independiente, su estimación de la media no es tan grande y hay una gran cantidad de variabilidad en ellos, la distribución exponencial debería ser una buena elección.

Debe evitarse la distribución normal porque sus valores están distribuidos simétricamente alrededor de la media y es una distribución sin límites, en la que se puede obtener un valor muy grande o muy pequeño, inclusive el caso de obtener valores negativos donde el simulador truncará ese valor a cero, causando que quizás se tome una acción no deseada.

Si no se conoce mucho acerca del proceso, pero se puede adquirir los valores mínimo y máximo se puede usar la distribución uniforme

En general, para seleccionar la distribución en ausencia de datos empíricos, primero deberá considerar las distribuciones exponencial, triangular y uniforme.

**Tabla 13. Posibles distribuciones en ausencia de datos**

DISTRIBUCIÓN	PARÁMETROS	CARACTERÍSTICAS	EJEMPLO DE USO
Exponencial	Media	Varianza alta Límite a la izquierda Sin límite a la derecha	Tiempo entre llegadas Tiempo para la falla de la máquina (índice de falla constante)
Triangular	Mínimo Moda Máximo	Simétrica o sin simetría Límite en ambos lados	Tiempos de actividad
Uniforme	Mínimo Máximo	Todos los valores igualmente parecidos Límite en ambos lados	Poco conocimiento del proceso

## 6. Documentar y aprobar los datos

Cuando se ha terminado de recolectar, analizar y convertir los datos es aconsejable organizarlos y documentarlos usando tablas, diagramas relacionales, listas con las suposiciones y las fuentes de los datos. El documento de datos no es estático y puede tener cambios en la vida del proyecto de simulación.

## EXPERTFIT

El software de simulación **Flexsim** incluye la aplicación **ExpertFit** para analizar y determinar el tipo de distribución de probabilidad de un conjunto de datos mediante pruebas de bondad de ajuste. Un objetivo secundario es proporcionar asistencia en el modelado de una fuente de aleatoriedad (por ejemplo, un tiempo de servicio) en ausencia de datos. Este software cuenta con los siguientes tipos de análisis:

- **Data Analysis:** se utiliza para determinar que distribución de probabilidad es la que mejor representa un conjunto de datos.
- **Task-Time Models:** módulo empleado para determinar la distribución de probabilidad para los tiempos de operación cuando no hay datos disponibles. Los datos requeridos son:
  - ✓ Tiempo mínimo de operación.
  - ✓ Tiempo de operación más probable (moda).
  - ✓ El percentil 100 o mayor valor del tiempo de operación.
- **Machine-Breakdown Models:** módulo utilizado para modelar los tiempos de parada aleatorios de una máquina cuando no hay datos de averías disponibles. Basado en estimaciones subjetivas de los parámetros tales como la eficiencia de la máquina (por ejemplo, 0,90) y media del tiempo de inactividad, ExpertFit especifica una distribución de tiempo ocupado y una distribución de tiempo de inactividad.

Adicionalmente los siguientes módulos están disponibles desde la opción menú:

- **Distribution Viewer:** usado para visualizar y calcular las características de las distribuciones sin tener que ingresar un conjunto de datos.
- **Batch Mode:** usado para ajustar distribuciones de uno o más conjuntos de datos con solo unos pocos clics.

**Tabla 14. Distribuciones de probabilidad disponibles en ExpertFit**

Continuous		Discrete
beta	log-Laplace	Bernoulli
Cauchy	log-logistic	binomial
chi-square	lognormal	geometric
Erlang	normal	hypergeometric
error	Pareto	logarithmic series
exponential	Pearson type 5	negative binomial
exponential power	Pearson type 6	Poisson
extreme value (2 types)	power function	uniform
F	random walk	
gamma	Rayleigh	
inverse Gaussian	Student's t	
inverted Weibull	triangular	
Johnson $S_B$	uniform	
Johnson $S_U$	Wald	
Laplace	Weibull	
logistic		

Hay 40 distribuciones de probabilidad disponibles en ExpertFit, de las cuales 8 son discretas y 32 continuas. Las distribuciones continuas están clasificadas en tres categorías:

- No negativas: 17 distribuciones.
- Sin límites: 10 distribuciones.
- Acotadas: 5 distribuciones.

ExpertFit permite exportar datos al siguiente software de simulación:

**Tabla 15. Software de simulación de eventos discretos soportado por ExpertFit**

AnyLogic	MedModel	Simio
Arena	Micro Saint	SIMPROCESS
AutoMod	OPNET Modeler	SIMUL8
AutoSched AP	ProcessModel	SLX
ExtendSim	ProModel	WITNESS
Flexsim	ServiceModel	

## METODOLOGÍA

La práctica consta de dos partes:

- Parte 1: correspondiente al ejercicio guía, que se acompaña de un archivo en ExpertFit con el desarrollo del ejemplo y un vídeo explicativo de la construcción del ejercicio.
- Parte 2: correspondiente a los ejercicios que el estudiante debe desarrollar.

## EJERCICIO GUÍA

Se utiliza el mismo ejercicio del Taller 1, con la diferencia de que se fabrican dos productos, con tiempos de proceso diferentes, para lo cual, se tiene un archivo en Excel con los tiempos de proceso de los productos.

Asumiendo que los objetos son los mismos del primer taller (ver figura 1), se comienza con crear dos tipos de producto diferentes, para lo cual, se crea en el *Trigger OnExit* de la FUENTE o del objeto Source, dos *itemtype* para representar a cada producto, tal como se observa en la figura 2.

Figura 1. Objetos del ejercicio guía

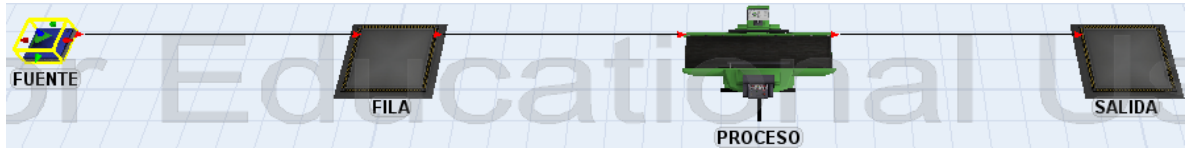
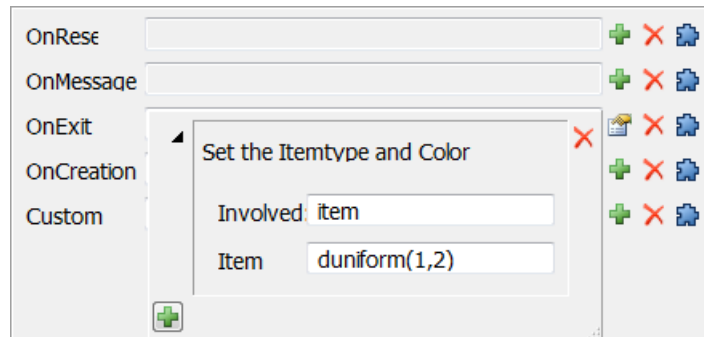


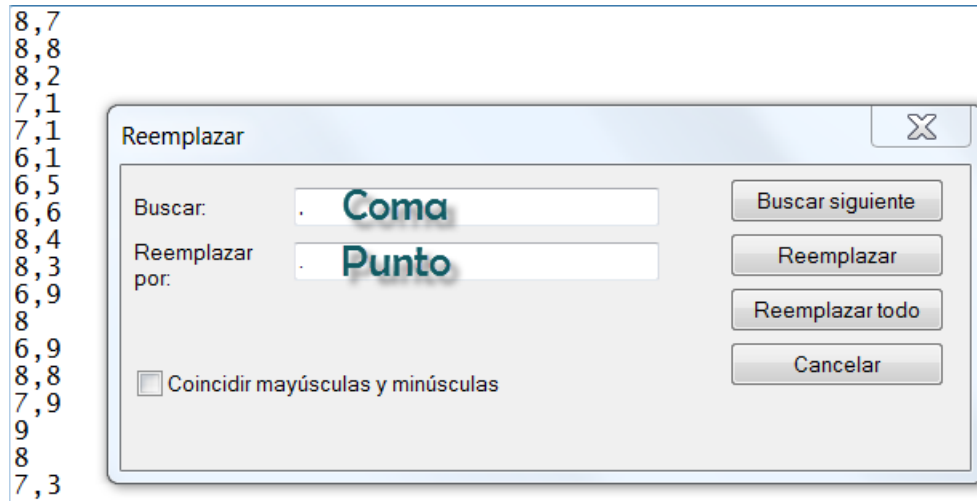
Figura 2. Trigger OnExit



A continuación, se ajustan los datos del tiempo de proceso para cada producto, que se encuentran en la hoja Ejercicio Guía del archivo Excel adjunto al taller con la herramienta ExpertFit de Flexsim.

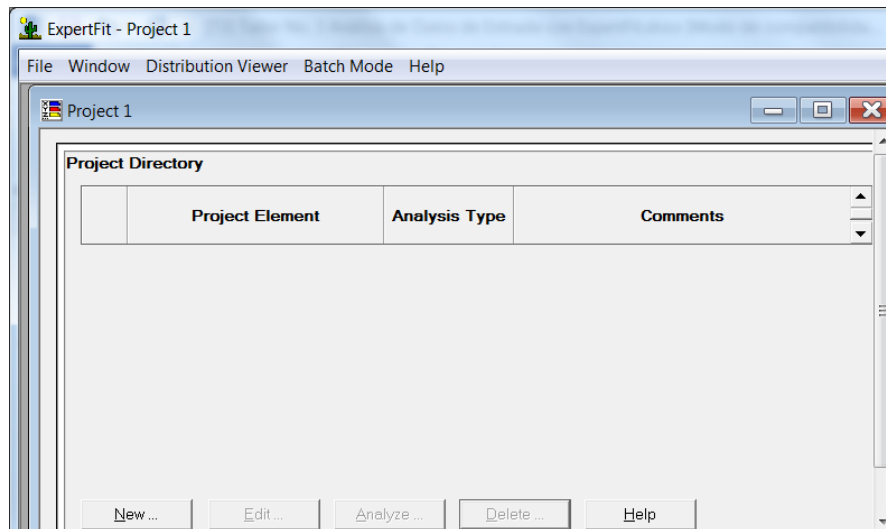
Una vez abierta la hoja de cálculo, se copian los datos del primer producto y se pegan en un bloc de notas, teniendo en cuenta que estos datos son reales y el separador de decimales es la coma, por lo cual, se debe reemplazar la coma por un punto en el bloc de notas donde se copiaron los datos, tal como se ilustra a continuación.

Figura 3. Cambiando el separador de decimales

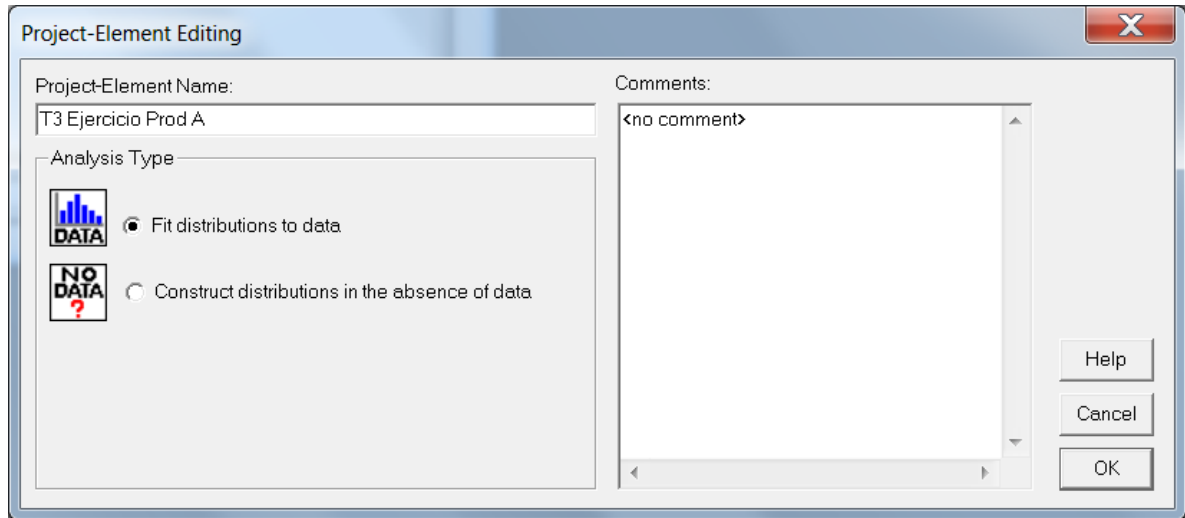


Una vez editados los datos, se guardan como un archivo de texto, en este caso con el nombre “Prod A.txt”. El mismo procedimiento se aplica a los datos del producto B. Posteriormente se abre el módulo ExpertFit que se encuentra en el menú *Statistics* de Flexsim. Una vez abierta la aplicación se siguen los pasos a continuación. Clic en New para crear un nuevo proyecto.

Figura 4. ExpertFit

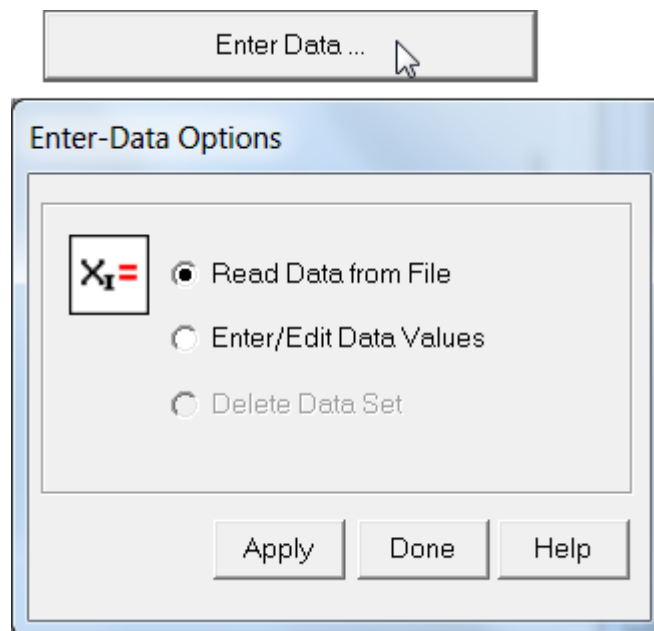


**Figura 5. Creando Proyecto de ExpertFit**



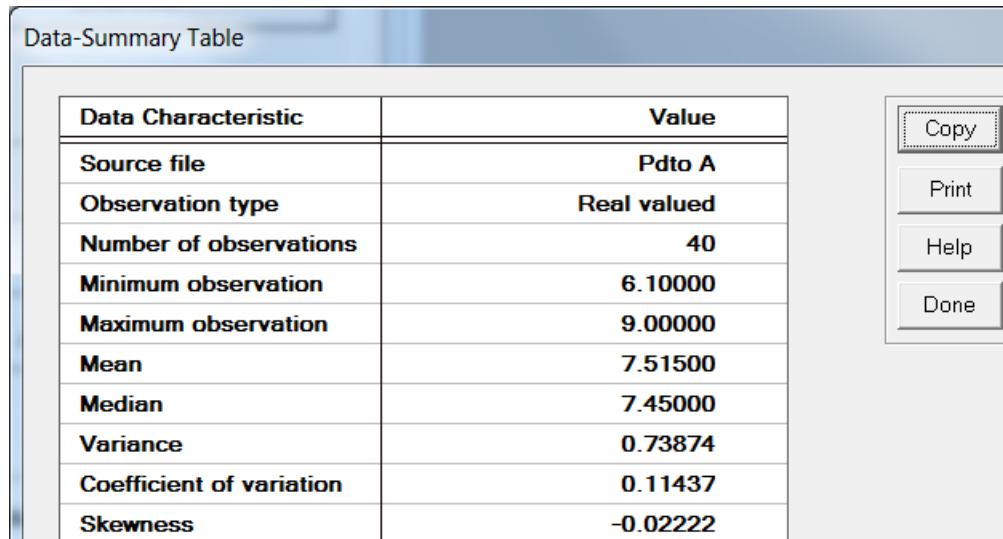
Se asigna un nombre al proyecto y se selecciona la primera opción que es para ajustar los datos a una distribución de probabilidad, desplegándose la siguiente ventana.

**Figura 6. Introduciendo datos**



En el tipo de archivo a abrir se cambia la extensión a texto y se escoge el archivo Prod A.txt, una vez escogido el archivo se presiona en el botón Apply y aparece una ventana que contiene las estadísticas descriptivas del conjunto de datos, tal como se aprecia a continuación.

**Figura 7. Resumen de los datos**



Data Characteristic	Value
Source file	Pdto A
Observation type	Real valued
Number of observations	40
Minimum observation	6.10000
Maximum observation	9.00000
Mean	7.51500
Median	7.45000
Variance	0.73874
Coefficient of variation	0.11437
Skewness	-0.02222

Presionando sobre el botón Done se regresa a la ventana principal donde se encuentra el botón *Histogram* que sirve para visualizar el histograma de los datos (ver figura 8).

Como se desea ajustar los datos de entrada, se presiona en la parte izquierda sobre Models y posteriormente sobre el botón Automated Fitting que como resultado muestra una ventana con las mejores distribuciones de probabilidad que se ajustan a los datos alimentados en ExpertFit (ver figura 9).

Figura 8. Histograma

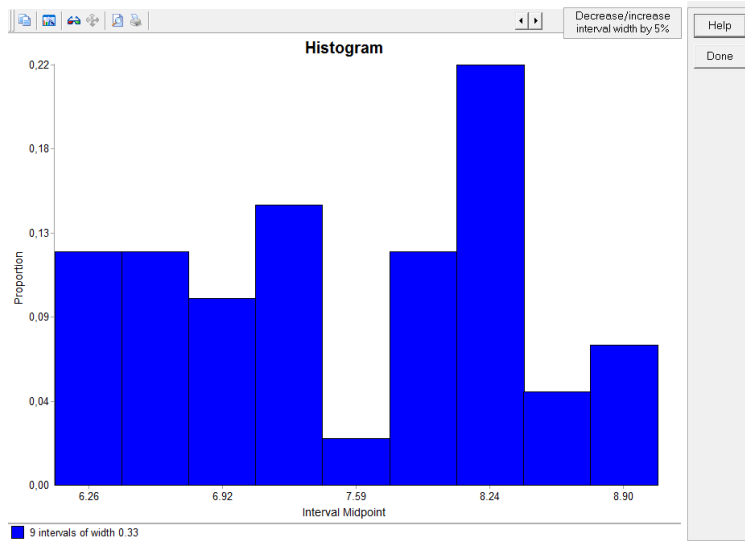


Figura 9. Distribuciones probables del conjunto de datos

Relative Evaluation of Candidate Models			
Model	Relative Score	Parameters	
1 - Johnson SB	98.39	Lower endpoint	6.03217
		Upper endpoint	9.04715
		Shape #1	0.03284
		Shape #2	0.62799
2 - Beta	95.97	Lower endpoint	6.08337
		Upper endpoint	9.01682
		Shape #1	0.97749
		Shape #2	1.03375
3 - Erlang(E)	84.68	Location	0.00171
		Scale	0.10435
		Shape	72

32 models are defined with scores between 0.00 and 98.39

---

**Absolute Evaluation of Model 1 - Johnson SB**

Evaluation: Good  
 Suggestion: Additional evaluations using Comparisons Tab might be informativ  
 See Help for more information.

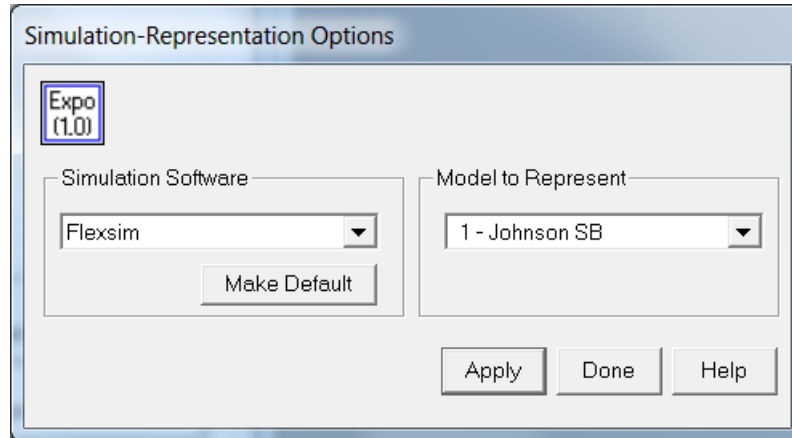
---

**Additional Information about Model 1 - Johnson SB**

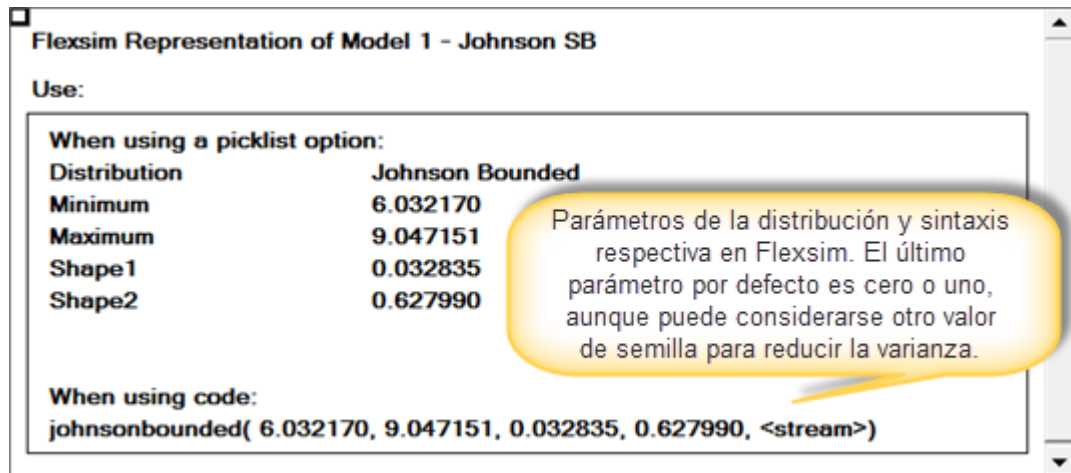
"Error" in the model mean  
 relative to the sample mean                      0.00240 = 0.03%

Como último queda escoger la distribución de probabilidad que más se ajuste a los datos y alimentarla en Flexsim.

**Figura 10. Representación de las distribuciones en Flexsim**

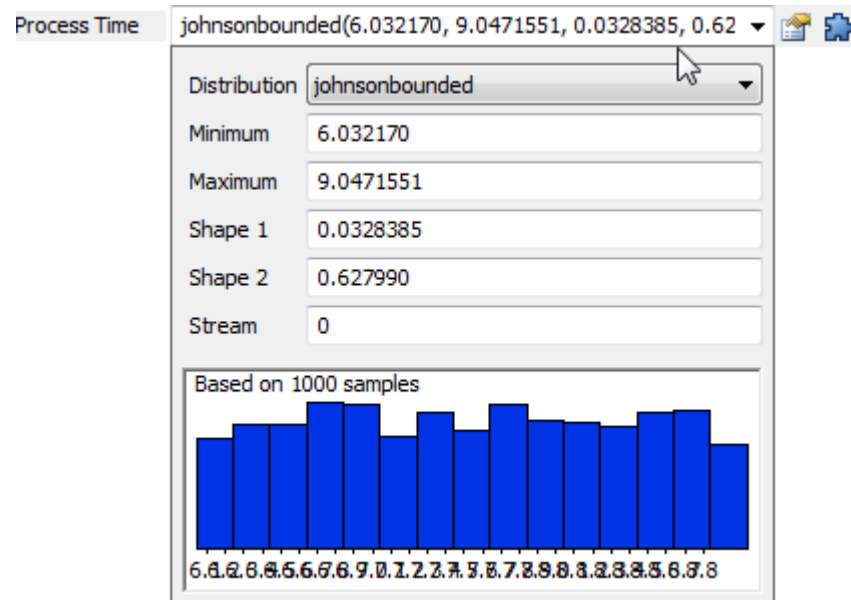


**Figura 11. Sintaxis en Flexsim**



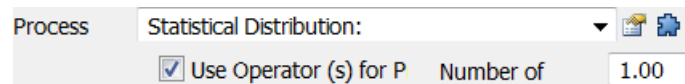
Una vez obtenidas la distribución de probabilidad, debe alimentarse en Flexsim en el objeto *Processor*, tal como se ilustra a continuación:

Figura 12. Distribución tiempo de proceso de producto A



Como son dos tiempos de ciclo para el *Processor*, uno para cada producto (A y B), se edita el código del tiempo de procesamiento:

Figura 13. Edición del tiempo de proceso



Y se reemplaza por las siguientes líneas:

```

1 treenode current = ownerobject(c);
2 treenode item = parnode(1);
3 /**Statistical Distribution: */
4 int type = getitemtype(item);
5
6 if (type==1)
7   return /**/exponential(0, 7, 0)/**/;
8 else if (type==2)
9   return /**/normal(9, 3, 0)/**/;

```

Introduciendo en cada *return* la distribución de probabilidad correspondiente con su respectiva sintaxis.

Para el desarrollo del taller, se introducen un nuevo objeto: *Combiner*, que se utiliza para combinar, ensamblar o hacer lotes de productos.

## ENSAMBLES

Cuando se necesita realizar un ensamble de dos o más entidades, se emplea el objeto *Combiner*, conectando las estaciones de donde provienen las entidades con el *Combiner*, tendiendo cuidado de realizar como primera conexión, la de la entidad base.

En la figura 14 se observa un *Combiner* llamado Ensamble, después del objeto WIP Ensamble de donde recibirá las entidades. En la figura 15 se observan las propiedades del objeto **Combiner**, donde se aprecia tipo de combinación a realizar: **Join** o Ensamble, la cual va a ensamblar 3 entidades que entran por el puerto 2, con una unidad del puerto 1, que es la entidad base y que no se aprecia en el área de componentes.

Figura 14. Combinación de Entidades

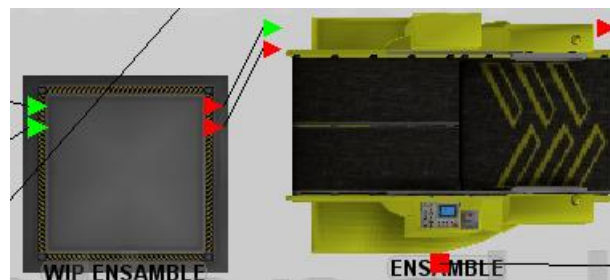
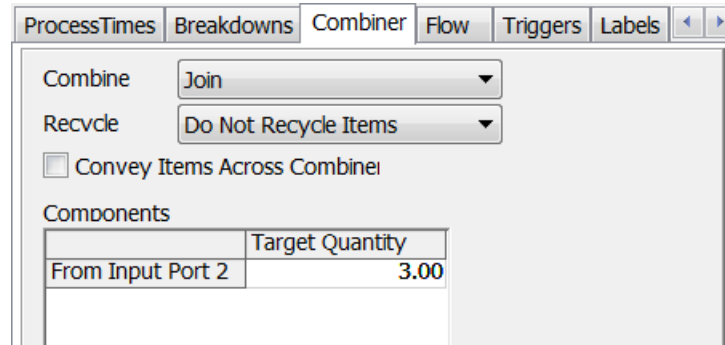


Figura 15. Sintaxis en Flexsim



## VISUAL TOOL COMO CONTENEDOR

Otro de los usos del objeto *VisualTool* es el de actuar como un contenedor de objetos, característica muy útil dado que se pueden utilizar submodelos o modelar áreas de un sistema en el que el *VisualTool* representa el layout de esa área y sus dimensiones representar las medidas reales del área en cuestión, los objetos contenidos representarían las estaciones, centros de trabajo o servidores de esa área, además al estar contenidos en otro objeto basta con desplazar el *VisualTool* para que se desplacen todos los elementos contenidos.

Los pasos a seguir son:

1. Agregar el *VisualTool* al layout del modelo.
2. Dimensionarlo: abrir pestaña *General* y editar el tamaño en X (SX) y el tamaño en Y (SY).
3. Arrastrar objetos dentro de el.
4. Desplazar el *VisualTool* contenedor para verificar que los objetos contenidos dentro de este objeto también se desplacen, manteniendo las mismas ubicaciones dentro del contenedor.

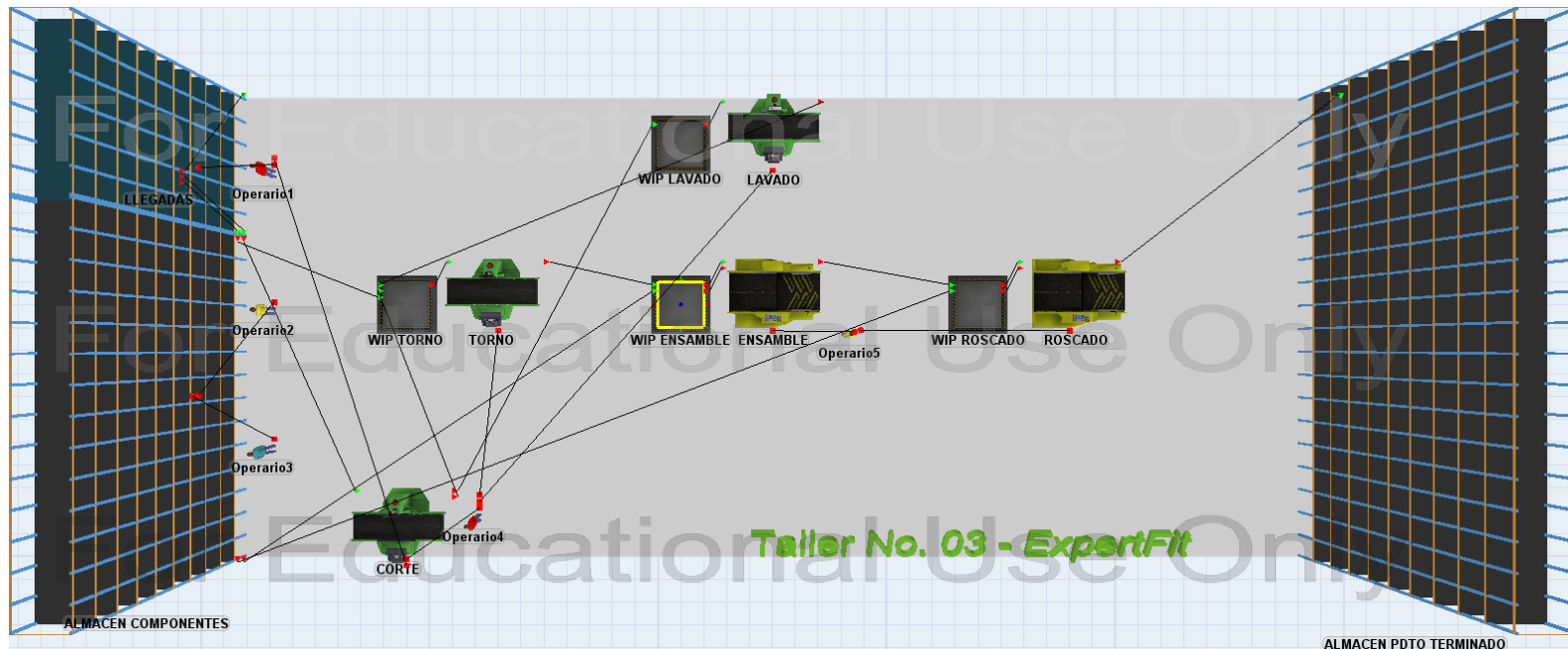
## TALLER A DESARROLLAR

### OBJETOS DEL MODELO

Tabla 16. Objetos del Taller No. 3

Elemento	Objeto	Nombre	Puertos	
			Entrada	Salida
Llegadas	Source	MATERIAL	0	1
		COMP1	0	1
		COMP2	0	1
Operaciones	Processor	CORTE	1	2
		LAVADO	1	1
		TORNO	1	1
	Combiner	ENSAMBLE	2	1
		ROSCADO	2	1
Filas	Queue	WIP LAVADO	1	1
		WIP TORNO	3	1
		WIP ENSAMBLE	2	2
		WIP ROSCADO	2	2
Almacenes	Rack	ALMACEN MATERIAL	1	2
		ALMACEN COMPONENTES	2	2
		ALMACEN PTERMINADO	1	0
Operarios	Operator	Operador1		
		Operador2		
		Operador3		
		Operador4		
		Operador5		

Figura 16. Esquema del Taller No. 3.



## DURACIÓN DEL TALLER

La duración estimada del taller es de 4 horas.

## DESCRIPCIÓN DEL TALLER

Simular en FLEXSIM una planta que elabora dos piezas:

- WI0406 que requiere 3 materias primas: 1 unidad de Material, 3 unidades de Comp2 y 2 unidades de Comp3.
- IW0315 que requiere una unidad de cada materia prima.

El proceso de producción se muestra en el diagrama de operaciones (ver figura 17).

Los materiales llegan a los almacenes de material y de componentes cada 24 horas en las siguientes cantidades:

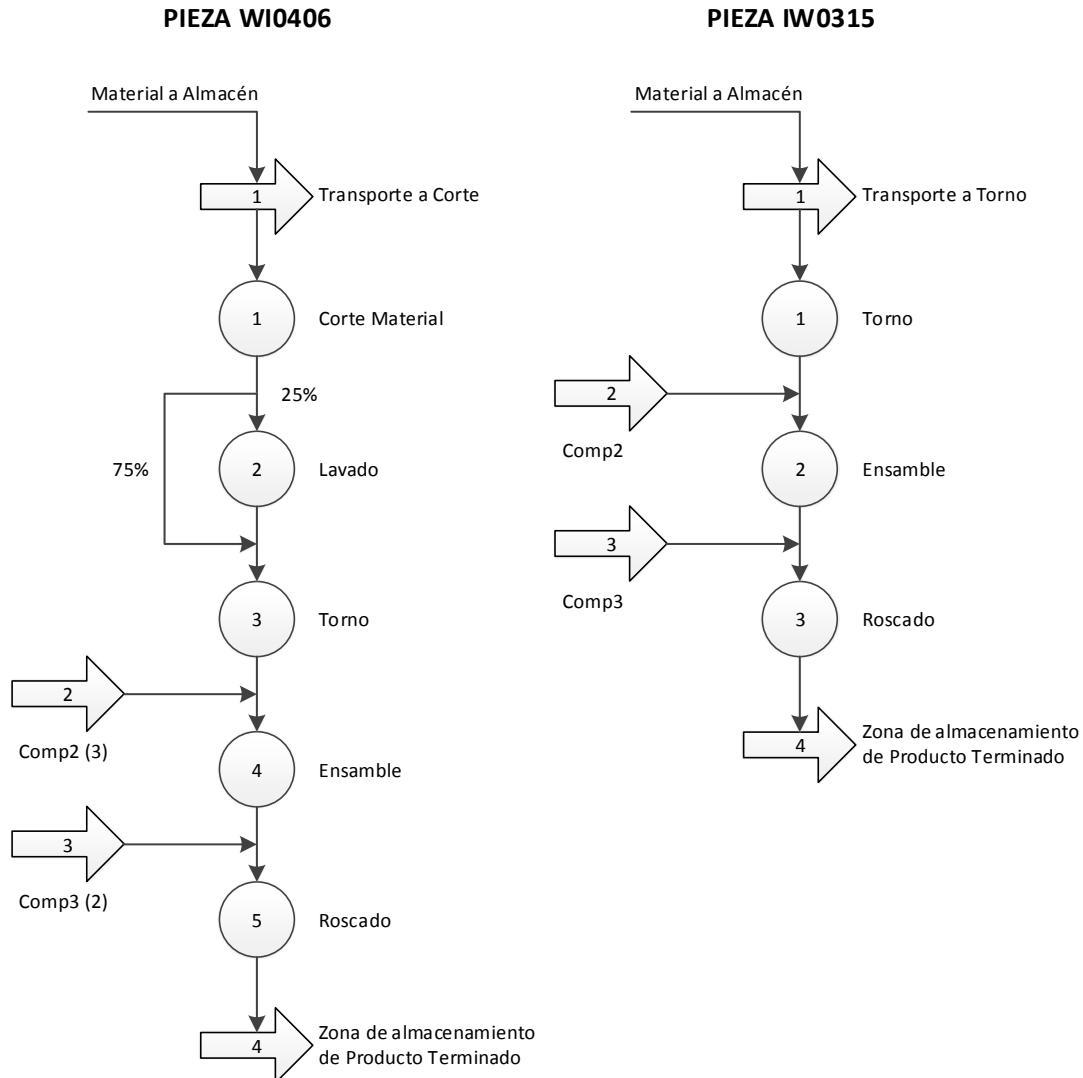
- Material: 40 unidades
- Componente 1: 80 unidades.
- Componente 2: 60 unidades.

Se utilizan 5 operarios para realizar el transporte de las materias primas a los centros de trabajo.

**Tabla 17. Secuencia de tareas de los operarios**

Operario	Secuencia
Operador1	ALMACÉN MATERIAL – CORTE – TORNO
Operador2	ALMACÉN COMPONENTES – ENSAMBLE
Operador3	ALMACÉN COMPONENTES – ROSCADO
Operador4	CORTE – LAVADO – TORNO – ENSAMBLE
Operador5	ENSAMBLE – ROSCADO – ALMACÉN PTERMINADO

Figura 17. Diagrama de operaciones



Para los tiempos de operación (minutos) se desarrolló un estudio de tiempos con 200 datos en cada centro de trabajo, discriminados por tipo de producto. En dos estaciones no se disponen de datos suficientes, solo se conocen el valor más probable y los valores mínimo y máximo.

Los datos de tiempos pueden consultarse en el archivo “Taller No. 3 ExpertFit - Tiempos”.

## ANÁLISIS DE DATOS

El análisis de los datos debe realizarse en la herramienta ExpertFit de Flexsim. Tenga en cuenta que si los datos no se ajustan a una distribución de probabilidad teórica conocida, puede utilizarse una distribución empírica (ver Taller No. 2).

## RECOMENDACIONES

- Se recomienda utilizar un *VisualTool* que actúe como contenedor de cada uno de los objetos del modelo. Este objeto se dimensiona y posteriormente se procede a agregar cada uno de los objetos que componen el sistema a simular.
- Cree el modelo para una sola pieza, posteriormente agregue el código necesario para la otra pieza.
- Se adjunta una presentación a este taller como guía, además del video incluido y un archivo en Excel con los datos de los tiempos de proceso en las estaciones.

## TALLER 4: SIMULACIÓN DE ESCENARIOS

### INTRODUCCIÓN

La utilización de escenarios de simulación para el análisis es una etapa que consiste en experimentar un modelo con diferentes valores en las variables de entrada en busca de las mejores condiciones de trabajo de los procesos que se simulan y evaluar diferentes alternativas de modelado.

Responde satisfactoriamente a preguntas hipotéticas del tipo “*qué pasa si...*”

### OBJETIVOS

- ❖ Experimentar en el modelo con diferentes escenarios de simulación, a través de la herramienta *Experimenter* de Flexsim.
- ❖ Conocer y aplicar las herramientas del software Flexsim para la solución de eventos relacionados con la teoría de colas.

## MARCO TEÓRICO

Uno de los aspectos interesantes de la simulación, es la posibilidad de evaluar diferentes situaciones futuras o alternativas de mejora al modelo desarrollado y ver el impacto de cada escenario de simulación para determinar las acciones a tomar. La evaluación de escenarios tiene las siguientes etapas:

- 1. Generación del modelo final.** Una vez que el modelo se ha validado, el analista está listo para realizar la simulación y estudiar el comportamiento del proceso. En caso de que se desee comparar escenarios diferentes para un mismo problema, éste será el modelo raíz; en tal situación, el siguiente paso es la definición de los escenarios a analizar.
- 2. Determinación de los escenarios para el análisis.** La utilización de escenarios de simulación para el análisis es una etapa de simulación que consiste en experimentar un modelo con diferentes valores en las variables de entrada. Cada escenario representa una configuración del modelo y responde satisfactoriamente a preguntas hipotéticas del tipo “*qué pasa si...*”.

Tras validar el modelo es necesario acordar con el cliente los escenarios que se quiere analizar. Una manera muy sencilla de determinarlos consiste en utilizar un escenario pesimista, uno optimista y uno intermedio para la variable de respuesta más importante. El analista también puede contribuir a la identificación y selección de los escenarios que se desean simular, tratando de reducir el número de combinaciones posibles de factores en la experimentación.

- 3. Análisis de sensibilidad.** Una vez que se obtienen los resultados de los escenarios es importante realizar pruebas estadísticas que permitan comparar los escenarios con los mejores resultados finales.

### Pasos para la implementación de escenarios:

1. Crear variables o etiquetas en los objetos de interés del experimento.
2. Definir las variables de análisis y los escenarios a simular con la herramienta *Experimenter*.
3. Utilizar código de programación en los objetos donde cambian los parámetros de entrada.
4. Realizar un análisis estadístico de las medidas de desempeño utilizadas con el fin de comparar las alternativas simuladas.

### METODOLOGÍA

La práctica consta de dos partes:

- Parte 1: correspondiente a un ejercicio guiado, junto con un vídeo explicativo de la construcción del ejercicio.
- Parte 2: correspondiente al ejercicio que el estudiante debe desarrollar.

### EJERCICIO GUÍA

Se utilizan los mismos objetos del ejercicio del Taller 3, tal como se aprecia en la figura 1.

Como ya se tiene un modelo preliminar, a continuación se definen los escenarios de simulación de interés del modelador o de los involucrados en la simulación.

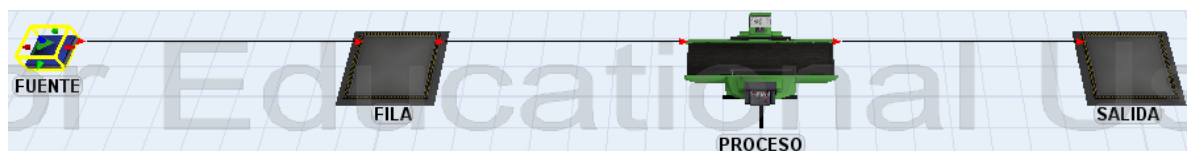
**Tabla 18. Escenarios para el objeto PROCESO**

Escenario	Objeto	Variable	Valor
1	PROCESO	Tiempo de proceso	10
2	PROCESO	Tiempo de proceso	30
3	PROCESO	Tiempo de proceso	50

**Tabla 19. Escenarios para el objeto FILA**

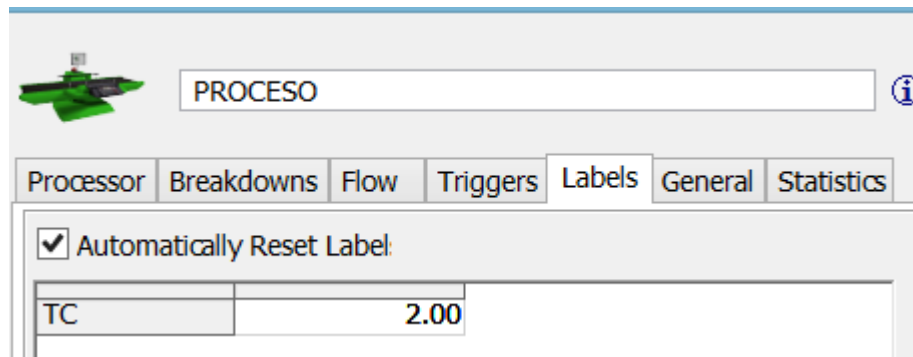
Escenario	Objeto	Variable	Valor
1	FILA	Máximo contenido	5
2	FILA		10
3	FILA		30

**Figura 1. Objetos del ejercicio guía**



Para el escenario del tiempo de proceso de la estación PROCESO, es necesario crear una etiqueta que gestione el escenario a correr, por lo cual, se crea la etiqueta numérica *TC* de la siguiente manera.

Figura 2. Etiqueta para controlar el escenario



Posteriormente se edita el código correspondiente al tiempo de proceso de la estación PROCESO.

```

1 treenode current = ownerobject(c);
2 treenode item = parnode(1);
3 /**By Expression*/
4 /** \nExpression: */
5 int type = /**/getlabelnum(current, "TC")/**/;
6 switch(type)
7 /** \n\nNote: The expression may be a constant
8 {
9   case 1: return 10;
10  case 2: return 30;
11  case 3: return 50;
12  default: return 100;
13 }

```

Se utilizan tiempos determinísticos para hacer más sencilla la tarea de verificar el modelo, sin embargo, bien podría usarse distribuciones probabilísticas. Una vez introducidos los tiempos para cada escenario, se abre la herramienta *Experimenter* de Flexsim, que se encuentra en el menú *Statistics*.

Se digitan los escenarios de simulación y las variables de control. Para el ejemplo se tienen 3 escenarios y 2 variables, como se observa en la figura a continuación y en el video adjunto al taller.

Figura 3. Herramienta Experimenter

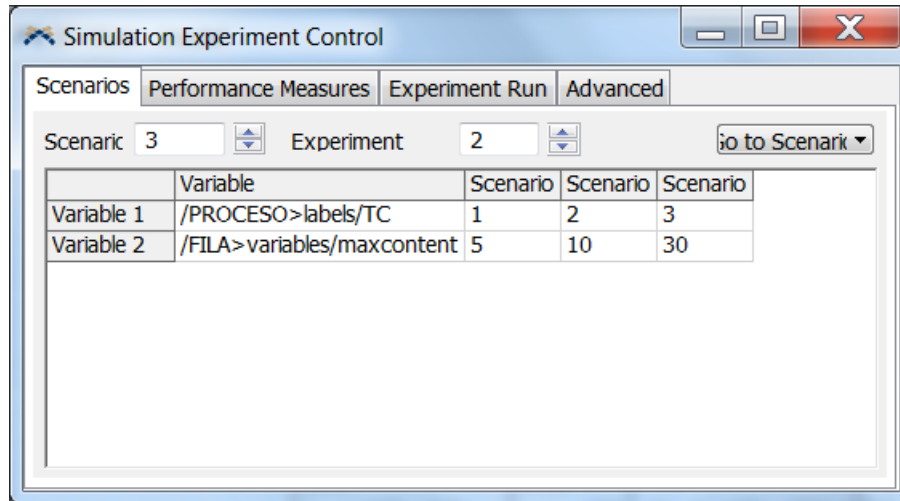
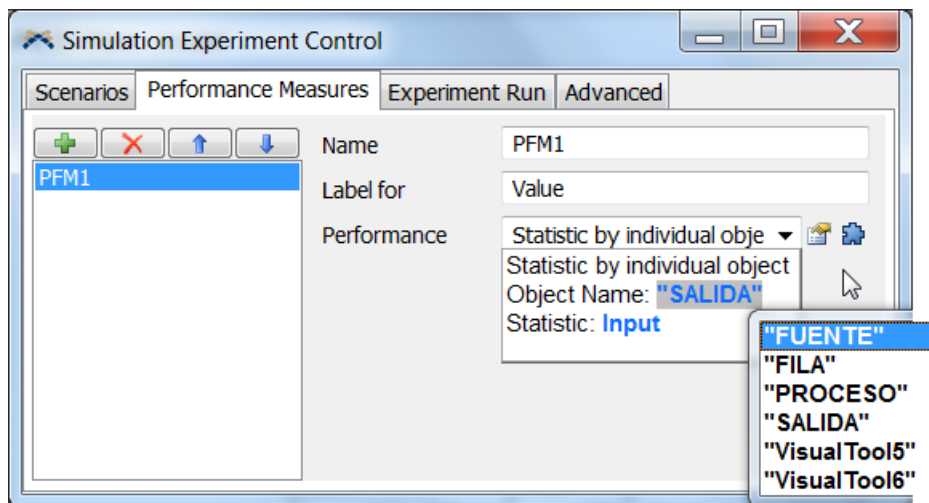


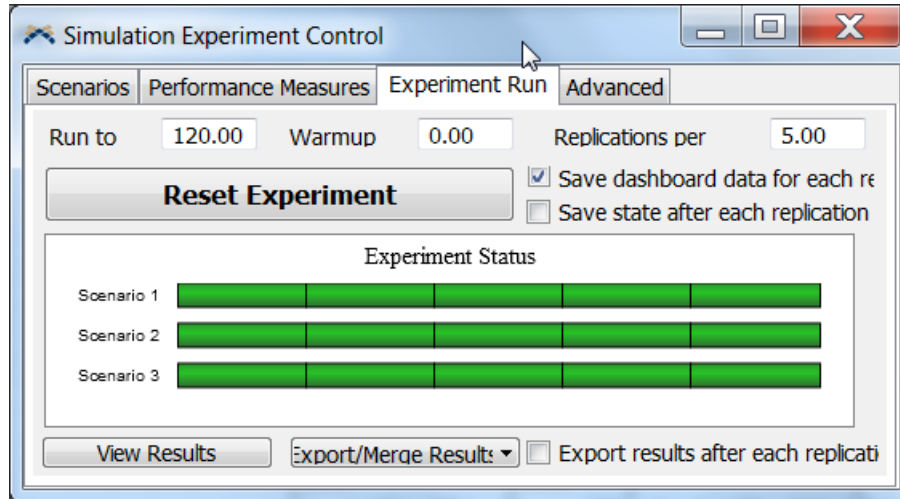
Figura 4. Medidas de Desempeño



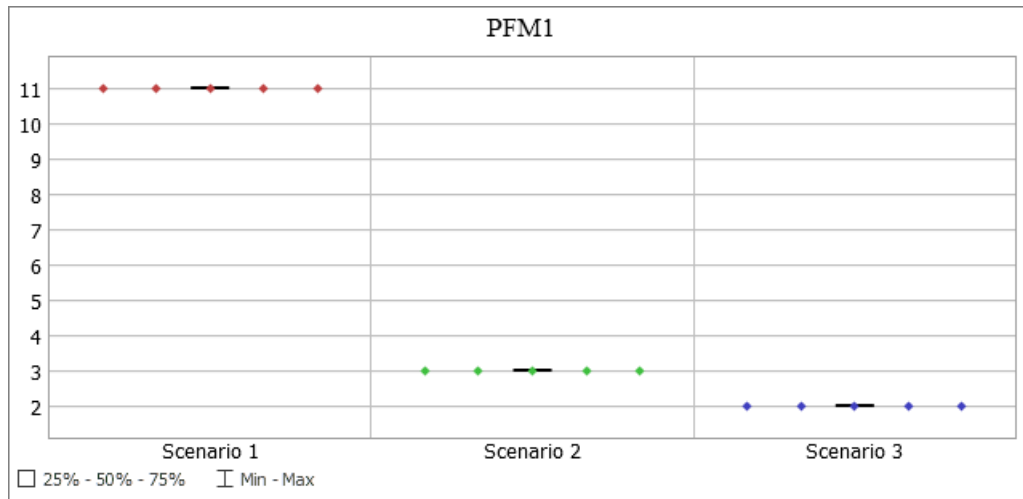
El siguiente paso consiste en definir las medidas de desempeño, tal como se observa en la figura anterior. Como medida de desempeño se toma el número de entidades que entran a la última estación llamada SALIDA. Por razones obvias, de antemano se sabe que el escenario con mejores resultados es el de menor tiempo de proceso o de operación y mayor contenido en el buffer de almacenamiento FILA, sin embargo, en escenarios con varios factores y más complejos esta

herramienta es útil para determinar la alternativa que arroja los mejores resultados o medidas de desempeño, combinando los factores.

**Figura 5. Corrida de los Escenarios**



**Figura 6. Resultados de la Medida de Desempeño**



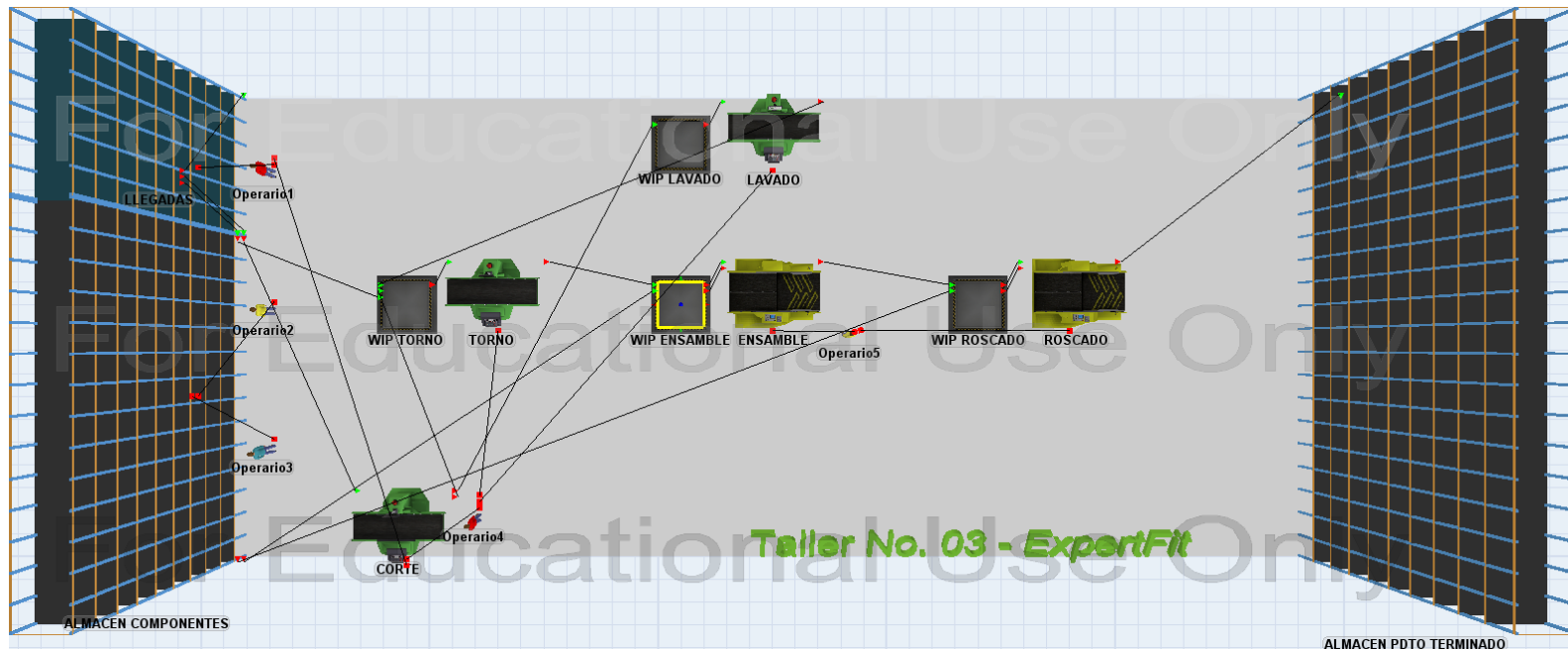
## TALLER A DESARROLLAR

### OBJETOS DEL MODELO

Tabla 20. Objetos del Taller No. 4

Elemento	Objeto	Nombre	Puertos	
			Entrada	Salida
Llegadas	Source	MATERIAL	0	1
		COMP1	0	1
		COMP2	0	1
Operaciones	Processor	CORTE	1	2
		LAVADO	1	1
		TORNO	1	1
	Combiner	ENSAMBLE	2	1
		ROSCADO	2	1
Filas	Queue	WIP LAVADO	1	1
		WIP TORNO	3	1
		WIP ENSAMBLE	2	2
		WIP ROSCADO	2	2
Almacenes	Rack	ALMACEN MATERIAL	1	2
		ALMACEN COMPONENTES	2	2
		ALMACEN PTERMINADO	1	0
Operarios	Operator	Operador1		
		Operador2		
		Operador3		
		Operador4		
		Operador5		

Figura 7. Esquema del Taller No. 4



## DURACIÓN DEL TALLER

La duración estimada del taller es de 4 horas.

## DESCRIPCIÓN DEL TALLER

Tomando como base el modelo del taller No. 3 (ver figura 8 Diagrama de operaciones), se desea simular los siguientes escenarios:

**Tabla 21. Escenarios para el objeto TORNO**

Escenario	Objeto	Variable	Valor
1	TORNO	Tiempo de proceso	exponential(0, 10, 1)
2	TORNO	Tiempo de proceso	normal(10, 1.5, 1)
3	TORNO	Tiempo de proceso	uniform(8, 12, 1)

**Tabla 22. Escenarios para el objeto CORTE y Operario4**

Escenario	Objeto	Variable	Valor
1	CORTE	Ubicación en el Layout	X: 7 Y: -17
	Operario4		X: 12 Y: -18
2	CORTE	Posición en X Posición en Y	X: 11 Y: -0
	Operario4		X: 16 Y: -1

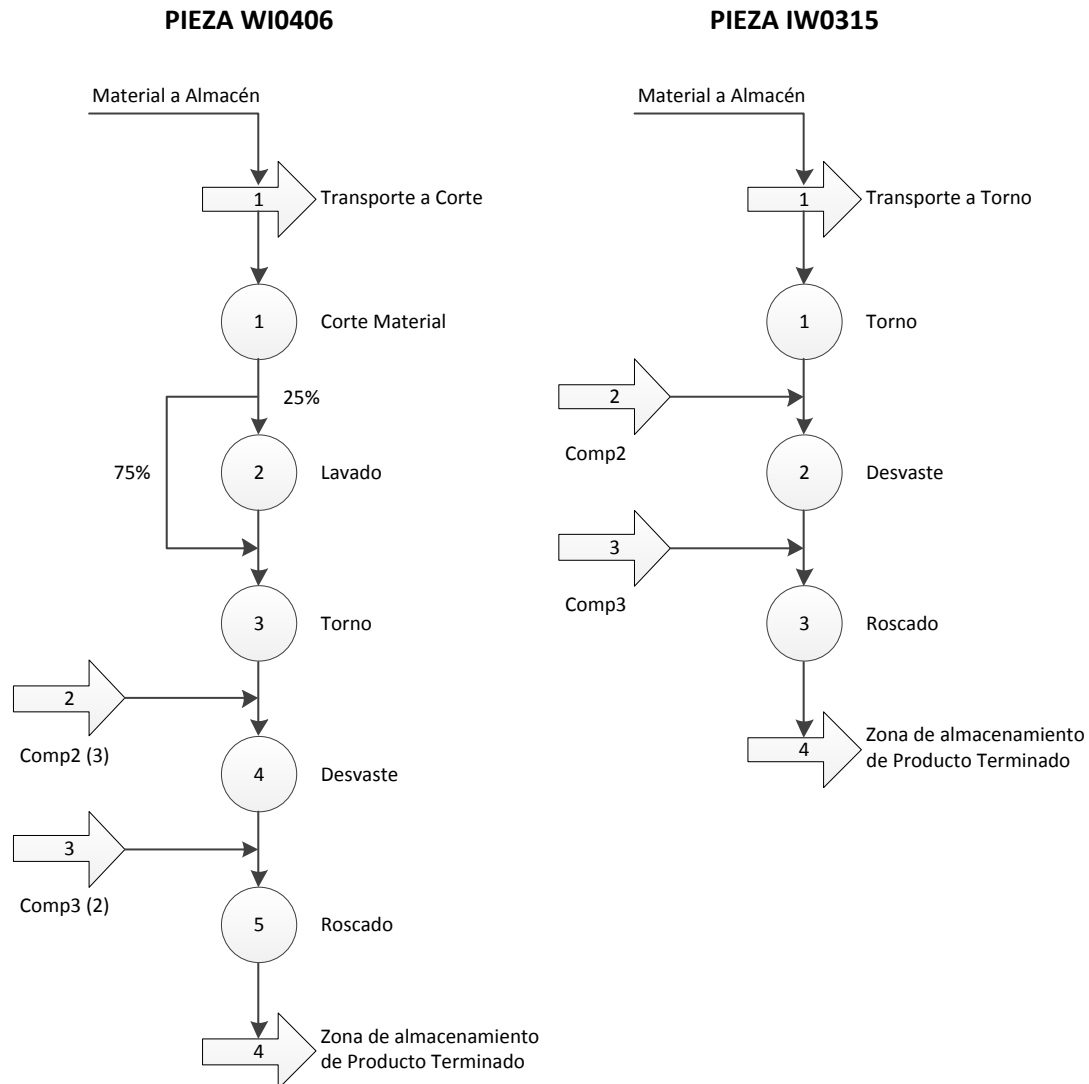
Se utilizan 5 operarios para realizar el transporte de las materias primas a los centros de trabajo.

**Tabla 23. Secuencia de tareas de los operarios**

Operario	Secuencia
Operador1	ALMACÉN MATERIAL – CORTE – TORNO
Operador2	ALMACÉN COMPONENTES – DESVASTE
Operador3	ALMACÉN COMPONENTES – ROSCADO
Operador4	CORTE – LAVADO – TORNO – DESVASTE
Operador5	DESVASTE – ROSCADO – ALMACÉN PTERMINADO

Tiempo de corrida de la simulación: 12 horas o 720 minutos.

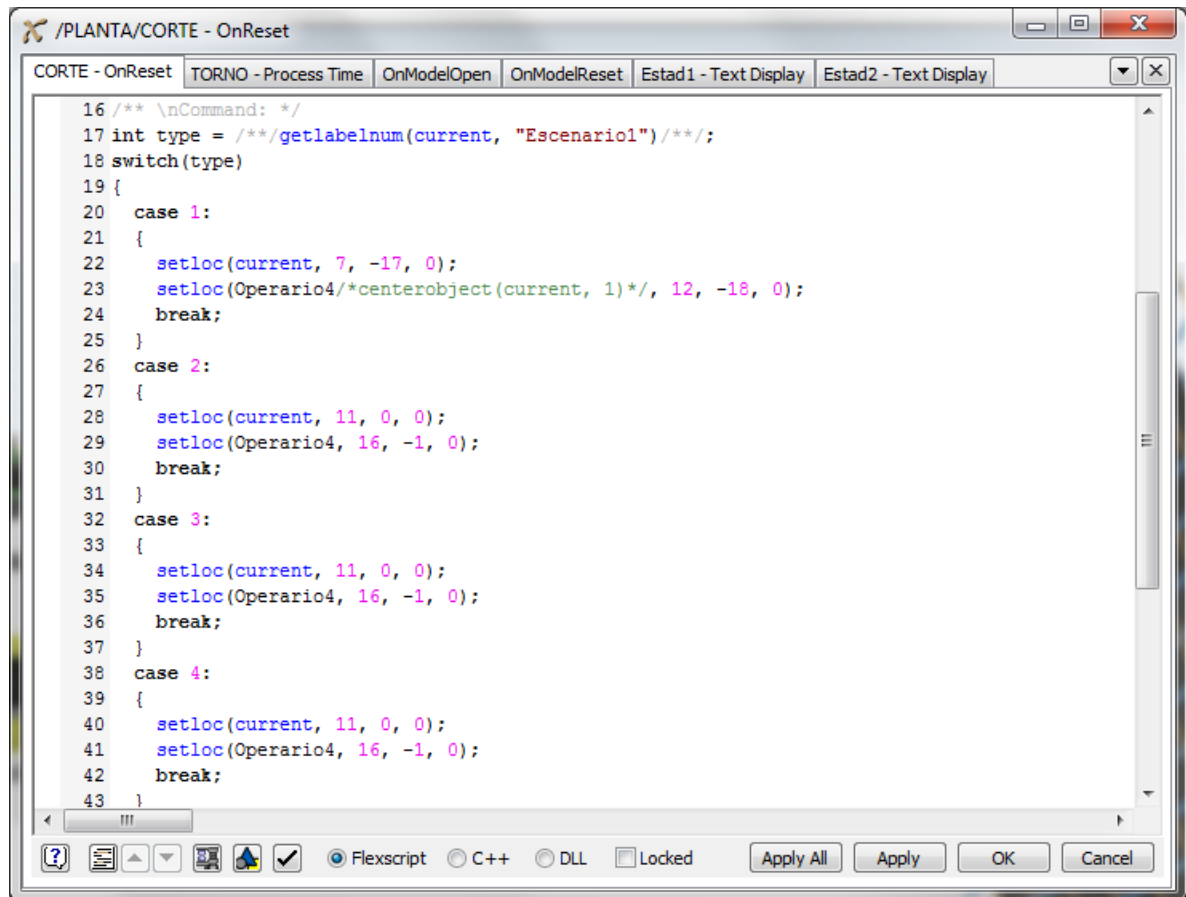
Figura 8. Diagrama de operaciones



## RECOMENDACIONES

- Utilizar etiquetas (labels) en los objetos que capturen el escenario a simular.
- Las variables del Experimenter deben ser numéricas.
- Emplear variables que midan el desempeño del escenario y del modelo para comparar las diferentes alternativas.

## CÓDIGOS



```

16 /** \nCommand: */
17 int type = /**/getlabelnum(current, "Escenario1")/**/;
18 switch(type)
19 {
20 case 1:
21 {
22     setloc(current, 7, -17, 0);
23     setloc(Operario4/*centerobject(current, 1)*/, 12, -18, 0);
24     break;
25 }
26 case 2:
27 {
28     setloc(current, 11, 0, 0);
29     setloc(Operario4, 16, -1, 0);
30     break;
31 }
32 case 3:
33 {
34     setloc(current, 11, 0, 0);
35     setloc(Operario4, 16, -1, 0);
36     break;
37 }
38 case 4:
39 {
40     setloc(current, 11, 0, 0);
41     setloc(Operario4, 16, -1, 0);
42     break;
43 }

```

Es necesario traducir los valores de los escenarios en códigos que permitan evaluar las alternativas deseadas.

Generalmente se suelen usar estructuras de decisión como:

- **If:** se utiliza para comparar dos o más valores y si es verdadero se ejecuta lo que está inmediatamente después, sino ejecuta el código después del **else**.

- **Switch:** es el típico case o estructura de selección, pero con la diferencia que se asigna bloque a cada una de las posibles alternativas que casi siempre son más de dos, por lo cual se prefiere sobre el if.

## NODOS

Para hacer referencia en el código a un objeto del modelo que no sea a través de los puertos de entrada o salida o a objetos que no estén relacionados, se puede utilizar el comando *node*.

Ejemplo:

```
treenode Operario4 = node("/PLANTA/Operario4", model());
```

Este comando crea un nodo llamado Operario4 que hace referencia al nodo del mismo nombre, utilizándose de manera similar a como se haría con el nodo *current* e *ítem*.

## TALLER 5: SISTEMAS DE CONTROL DE PRODUCCIÓN

### INTRODUCCIÓN

Un sistema de control de producción efectivo es aquel que produce las partes correctas, en el momento correcto y a un costo competitivo (Huang, Wang y Pi, 1998). Existen numerosos sistemas de producción, sin embargo, todas sus estrategias de control se dividen en *pull* (halar) y *push* (empujar) o en sistemas híbridos que los combinan.

### OBJETIVOS

- Simular un sistema de producción CONWIP y compararlo con un sistema push normal donde no se controla el inventario en proceso.
- Conocer la utilización de los TaskSequence.

### MARCO TEÓRICO

Un sistema push programa la ejecución de trabajos basado en la demanda (pronóstico), mientras que un sistema pull autoriza la ejecución de los trabajos basado en el control de los inventarios. Como consecuencia, los sistemas push controlan el *throughput* y observan el trabajo en proceso, mientras que los sistemas pull controlan el trabajo en proceso y miden el *throughput* (Hopp y Spearman, 1996).

Dentro de las estrategias de control de producción existen algunas ampliamente difundidas como el sistema Kanban y el MRP II, y otras menos conocidas pero igualmente importantes, como el CONWIP.

El sistema de producción CONWIP recibe su nombre del acrónimo en inglés para CONstant Work in Process, que traduce “trabajo en proceso constante”. Este concepto de sistemas pull fue introducido por Spearman, Woodruff y Hopp (1990) y es un sistema Kanban mejorado, por lo tanto, su funcionamiento es mejor cuando el flujo es uniforme, aunque puede utilizarse en varios entornos de manufactura.

Los sistemas pull establecen una cantidad máxima de trabajo en proceso suficiente para alcanzar el throughput (TH) deseado, pero este trabajo en proceso también debe ser lo suficientemente pequeño como para evitar tiempos de ciclo demasiado grandes (debidos al exceso). La forma más simple de alcanzar este objetivo es por medio del CONWIP, el cual establece un límite para el trabajo en proceso y lo mantiene constante, sincronizando la ejecución de los trabajos al sistema de producción con la terminación de ellos. De este modo, cada vez que un trabajo sale del sistema se autoriza el ingreso de uno nuevo.

El control en un sistema CONWIP se realiza con señales que pueden ser tarjetas o contenedores. Las tarjetas se asignan a la línea de producción en lugar de a un producto específico y se fijan a los contenedores al principio de la línea en donde se autoriza el ingreso de un trabajo que se encuentra en la lista de faltantes (que proviene de un programa maestro de producción o que se ha obtenido mediante algún otro procedimiento) y para el cual hay materia prima disponible.

La tarjeta viaja con el contenedor hasta el final de la línea en donde se retira y regresa al inicio, ingresando a una fila de espera o siendo fijada a un nuevo trabajo.

Un contenedor entrará a la línea solamente cuando exista una tarjeta disponible, de lo contrario ningún trabajo será ejecutado, aunque la primera estación de trabajo se encuentre inactiva.

De este modo las tarjetas de producción cumplen la función de mantener constante el trabajo en proceso y de decidir cuándo ingresan los trabajos a la línea, mientras que la lista de faltantes decide qué trabajos entran en la línea y en qué secuencia.

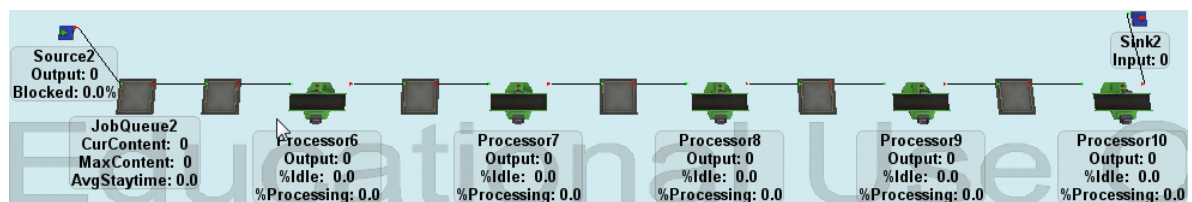
## EJERCICIO GUÍA

Implementar una línea de producción compuesta por 5 estaciones y 5 buffer de almacenamiento o colas antes de las estaciones. Además incluya una cola adicional al inicio de la línea de producción con la cual se controlara el WIP.

Los tiempos de ciclo están distribuidos exponencialmente con una media de 10 minutos para todas las estaciones, a excepción de la última estación que tendrá una media de 20 minutos y representará el cuello de botella de la línea. El tiempo entre llegadas en la fuente es de 10 minutos distribuidos exponencialmente.

El layout del sistema se observa en la figura 1.

**Figura 9. Layout del modelo**



Una vez introducidos los objetos del modelo, se seleccionan todos para copiarlos y pegarlos en una nueva ubicación, paralela al modelo inicial. Esto se hace con el fin de comparar los modelos visualmente, uno frente al otro.

### ¿Cómo implementar un sistema CONWIP en Flexsim?

Se crean 3 variables globales con las que se controlan los niveles de inventario de los sistemas y la restricción de inventario en proceso.

**Tabla 24. Variables globales del modelo**

Variable Global	Tipo de variable	Descripción
<b>WIP</b>	Integer	Work In Process para el sistema CONWIPi
<b>WIPlimit</b>	Integer	Restricción o límite de inventario del sistema CONWIP
<b>WIP2</b>	Integer	Work In Process para el sistema WIP (Push)

Fuente: Autor.

En la cola de entrada a la línea de producción hay un objeto llamado JobQueue, en el trigger OnExit se debe incrementar el inventario en proceso de cualquiera de los dos sistemas, de la siguiente manera:

```
1 /**Custom Code*/
2 treenode item = parnode(1);
3 treenode current = ownerobject(c);
4 int port = parval(2);
5
6 WIP2++;
7 colorarray(item, 2);
```

La instrucción de la línea 6 es la que ocasiona que se incremente dicha variable, correspondiente al trigger OnExit del objeto JobQueue de los dos sistemas. Posteriormente cuando las entidades entren al objeto Sink o salgan del sistema, deb ser decrementada en una unidad las variables del inventario en proceso, de la siguiente forma:

```
1 /**Custom Code*/
2 treenode item = parnode(1);
3 treenode current = ownerobject(c);
4 int port = parval(2);
5
6 WIP2--;
7
```

Cambiando únicamente los operadores matemáticos de la instrucción de la línea 6.

La diferencia entre un sistema y otro radica en comparar los niveles de inventario, que se realizan al principio de la línea en el objeto JobQueue.

```
1 /**Custom Code*/
2 treenode item = parnode(1);
3 treenode current = ownerobject(c);
4 int port = parval(2);
5
6 WIP++;
7 colorarray(item, 7);
8 if (WIP>=WIPlimit)
9     closeoutput(current);
```

Si el inventario actual es mayor al límite establecido, se cierra el puerto de salida de JobQueue. Ahora cuando una entidad es retirada o sale del sistema, el inventario disminuye, por lo tanto, es necesario abrir el puerto de salida de la estación JobQueue a cargo de controlar el inventario. Esta operación se realiza en el Sink ubicado antes del cuello de botella.

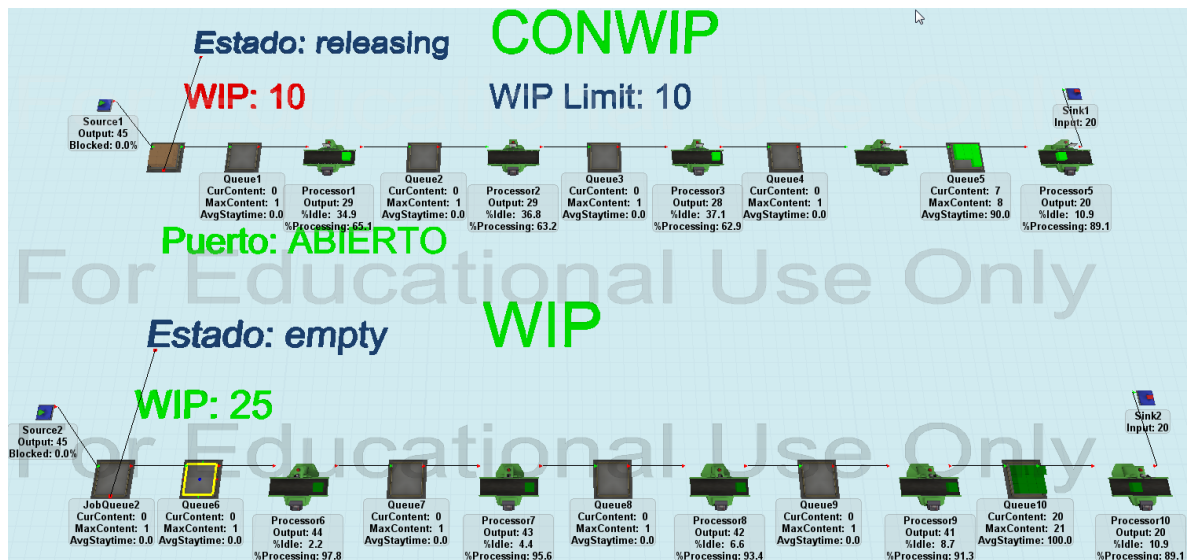
```

1 /**Custom Code*/
2 treenode item = parnode(1);
3 treenode current = ownerobject(c);
4 int port = parval(2);
5
6 WIP--;
7 openoutput(JobQueue);

```

Ahora estando todo listo, se simula por un período de tiempo determinado, por ejemplo, simular por un tiempo de 480 minutos y comparar los dos modelos. Esta comparación se observa en la figura 2.

Figura 10. CONWIP vs WIP (Push)



## TALLER 6: PROYECTO INTEGRADOR

### INTRODUCCIÓN

Esta práctica integra varios de los conceptos y herramientas vistas en talleres anteriores como el análisis de datos de entrada, redes de transporte, flujo multiproducto, importación de layout con dimensiones y escenarios de simulación.

### OBJETIVOS

- Desarrollar un modelo de un sistema real o abstracto que integre los elementos vistos en talleres anteriores.
- Proponer una alternativa de mejora y simularla junto con la situación actual.

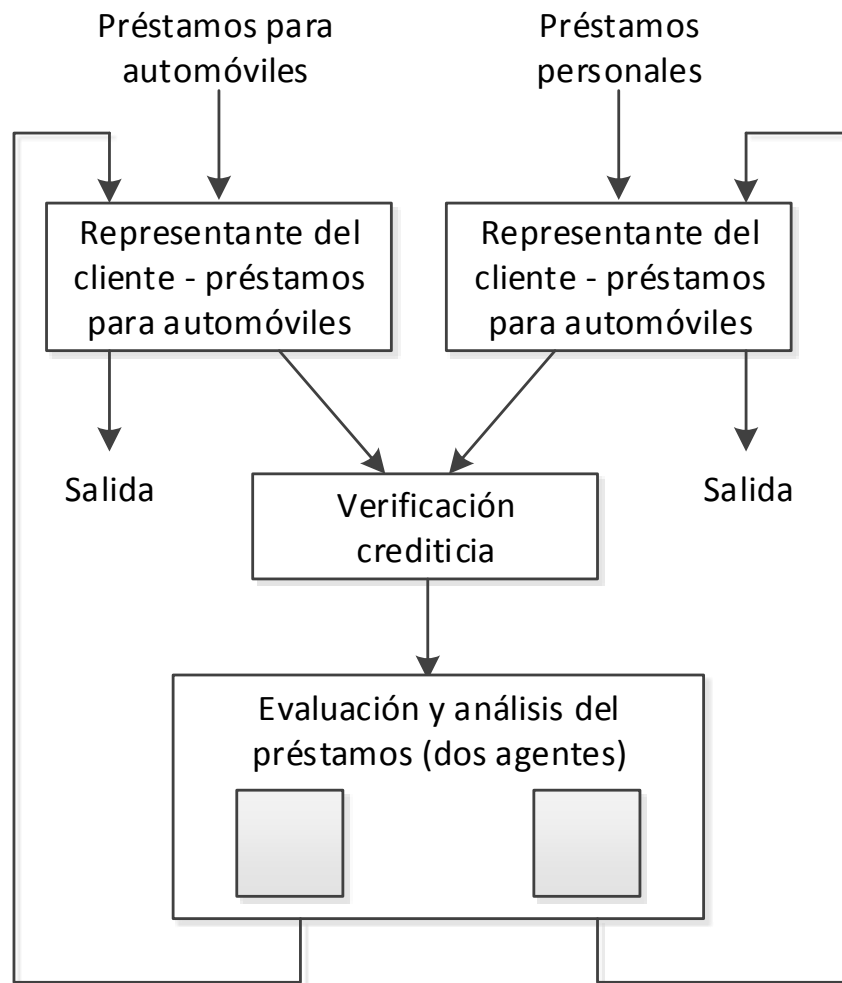
### EJERCICIO GUÍA

### DESCRIPCIÓN

Un banco estableció el siguiente sistema de procesamiento para préstamos automóviles y para préstamos personales de otra índole. Ver figura 1. Un representante de servicio al cliente se especializa en préstamos para automóviles y cuando un cliente llega o llama se le envía a ese lugar; otro representante maneja otro tipo de préstamos personales.

Estos representantes ayudan al cliente a preparar las solicitudes necesarias. Luego, las revisan para asegurarse que están completas. Cada representante maneja solo el tipo particular de préstamos que le corresponde (automóviles frente a otros préstamos personales), y si un cliente llega cuando aquel se encuentra ocupado, se le pide que espera.

**Figura 1. Diagrama de flujo del sistema de créditos**



Después de que las solicitudes han sido revisadas por el representante del cliente, se envían a la sección de revisión de crédito, atendida por una persona, quien reúne la historia crediticia del solicitante, a partir de los registros y de agencias de crédito. Las solicitudes son procesadas en orden PEPS.

Cuando se termina la etapa de revisión del crédito, las solicitudes pasan al departamento de análisis y evaluación de préstamos, donde se toma una decisión final. Este departamento es atendido por dos funcionarios que manejan cada uno un tipo de solicitud, las cuales también se procesan aquí en orden PEPS.

Finalmente, las solicitudes se devuelven al representante del cliente, de manera que pueda informarse a los clientes acerca de la decisión y puedan suministrarse las firmas finales si es necesario. Observar que el representante del cliente maneja dos tareas diferentes; suponer que ambas se realizan con un orden PEPS.

Considerar que las solicitudes de préstamos para automóviles llegan al escritorio del representante con 80 minutos de diferencia en promedio. El representante emplea un promedio de 43 minutos para procesar la primera etapa del préstamo y 20 minutos adicionales, en promedio, para procesar la segunda etapa (información acerca de la decisión y firmas). Para otros préstamos personales, la media del tiempo entre llegadas es de 90 minutos, y el representante de servicio al cliente toma 60 minutos para procesar la primera etapa y 15 para la segunda en promedio.

Para ambos tipos de préstamos, se requiere un promedio de 26 minutos para hacer la revisión del crédito y 76 minutos para hacer la evaluación y el análisis del préstamo.

Todos estos tiempos son promedio, y los tiempos reales varían considerablemente del promedio, dependiendo de la complejidad del préstamo y del historial crediticio del solicitante.

El propósito de este problema es aplicar alguna de las perspectivas acerca de sistemas de procesamiento en el gran tema de las teorías de colas.

- a. ¿Dónde están las situaciones más riesgosas de este sistema?, es decir, en qué estación (es) habrá acumulación de solicitudes y congestión?
- b. ¿Qué sugerencias podrían hacerse para mejorar la operación de este sistema? ¿Cuál es la base para estas sugerencias?

Fuente: Análisis Cuantitativo para los negocios. Bonini, Hausman y Bierman. McGraw Hill.

Figura 2. Objetos del ejercicio del taller 6

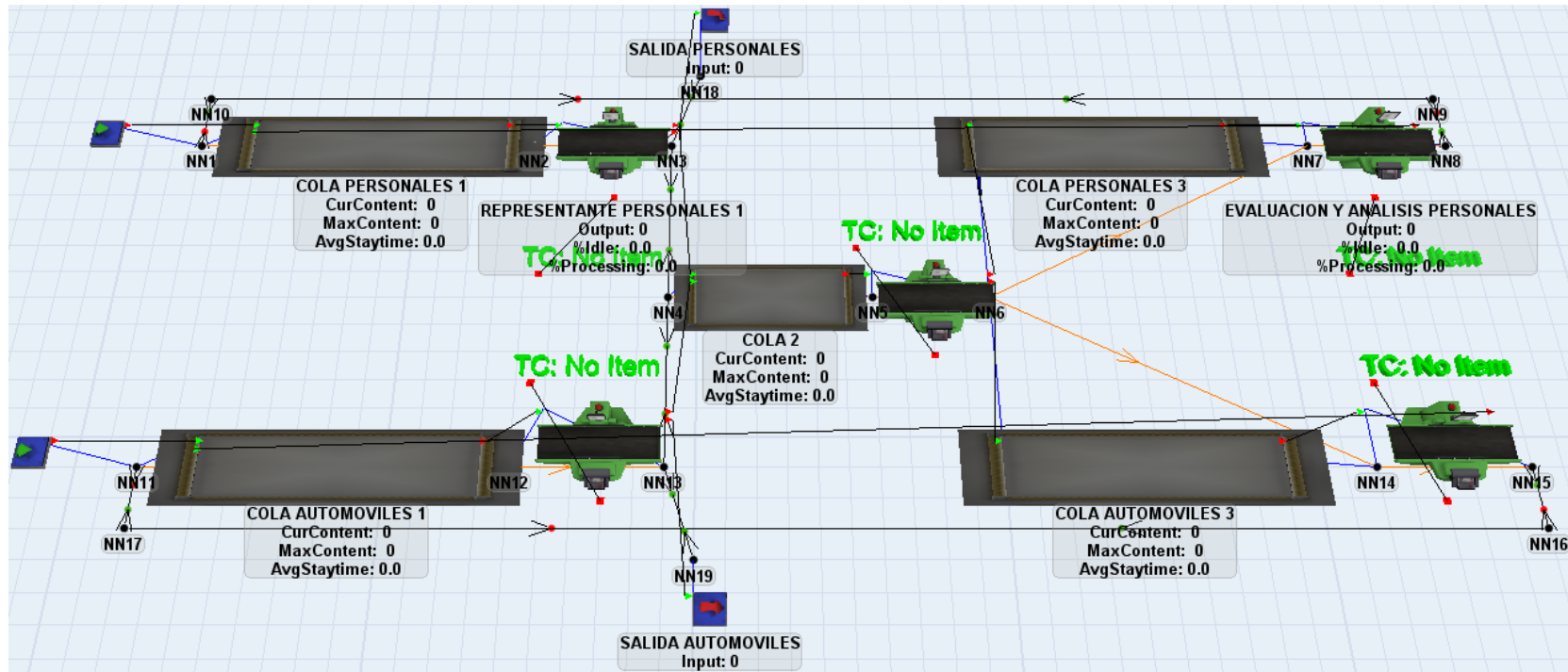


Tabla 1. Elementos del modelo

Elemento	Objeto	Nombre	Puertos		
			Entrada	Central	Salida
Llegadas	Source	CREDITOS PERDONALES	0	0	1
		CREDITOS AUTOMOVILES	0	0	1
Filas	Queue	COLA PERSONALES 1	1	1	1
		COLA AUTOMOVILES 1	1	1	1
		COLA 2	1	1	1
		COLA PERSONALES 3	1	1	1
		COLA AUTOMOVILES 3	1	1	1
Operaciones	Processor	REPRESENTANTE PERSONALES 1	1	1	1
		REPRESENTANTE AUTOMOVILES 1	1	1	1
		VERIFICACION CREDITICIA	2	1	1
		EVALUACION Y ANALISIS PERSONALES	1	1	1
		EVALUACION Y ANALISIS AUTOMOVILES 1	1	1	1
Salidas	Sink	SALIDA PERSONALES	1	0	0
		SALIDA AUTOMOVILES	1	0	0

Fuente: Autor.

## BIBLIOGRAFÍA

- GARCÍA, Francisco; SIERRA, Jorge y GUZMÁN Virginia. Simulación de Sistemas para Administración e Ingeniería. CECSA, México, 2005.
- GUASH, Antoni; PIERA, Miguel Ángel; CASANOVAS, Josep y FIGUERAS, Jaume. Modelado y simulación. Aplicación y procesos logísticos de fabricación y servicios. Alfaomega, México, 2005.
- GARCÍA, Eduardo; GARCÍA, Heriberto y CARDENAS, Leopoldo. Simulación y análisis de sistemas con ProModel. Pearson Educación, México, 2006.
- BLANCO Rivero, Luis; FAJARDO Piedrahita, Iván. Simulación con ProModel: casos de producción y logística. Escuela Colombiana de Ingeniería, Bogotá, 2003.
- GARAVITO Hernández, Edwin A.; ARENAS Díaz Piedad. Simulación de Procesos de Manufactura 1. Publicaciones UIS, Bucaramanga, 2008
- HARREL, Charles; GHOSH Biman. Simulation using ProModel. First Edition, McGraw Hill, United States of America, 2006.
- KELTON, W. David; SADOWSKI, Randall and SADOWSKI, Deborah. Simulation with Arena. Second Edition, McGraw Hill, United States of America, 2000.

- Portal Web de la Winter Simulation Conference:  
<http://www.wintersim.org>
- Portal Web de Vatic Consulting Group:  
<http://www.vaticgroup.com/unlimitpages.asp?id=19>
- Página Web de Flexsim Simulation Software:  
<http://www.flexsim.com/>
- Página Web de Flexsim South America  
<http://www.flexsim.cl/>
- Página Web de Flexsim Entrenamiento Online:  
<http://www.competition.flexsim.com/entrenamiento/>
- Sitio Web de ProModel Corporation:  
<http://www.promodel.com/>

**ANEXO C**  
**Encuesta de Evaluación de los Talleres**



ENCUESTA DE EVALUACIÓN

Estudiante:	Leonardo Camargo Restrepo	Código:	2063421
Asignatura:	Técnicas Modernas de Optimización	Código:	23536
Docente:	Ing. Edwin Alberto Garavito Hernández	Fecha:	09 / 04 / 2012
Práctica:	Taller No. 3 – Análisis de Datos con ExperFit		

Por favor diligencie este formato, teniendo en cuenta la escala de 1 a 5. Marque con una X la calificación correspondiente teniendo en cuenta los siguientes criterios:

1. Deficiente	2. Regular	3. Bueno	4. Muy bueno	5. Excelente
---------------	------------	----------	--------------	--------------

1. PRESENTACION

Descripción	Calificación				
	1	2	3	4	5
La presentación general del material facilita su lectura y utilización como guía de estudio.					X
Las imágenes contenidas en el taller facilitan la comprensión de los conceptos estudiados en el mismo.					X
La proporción, tamaño y distribución del taller es adecuado.					X
Las instrucciones que se imparten son claras y fáciles de seguir.				X	
El taller esta estructurado de tal manera que contribuye a la comprensión del software.					X



## 2. LENGUAJE

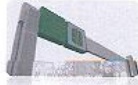
Descripción	Calificación				
	1	2	3	4	5
El taller es explícito y contiene los elementos necesarios para realizar la práctica.					X
El lenguaje utilizado es claro, comprensible y se ajusta a mis conocimientos.				X	
El taller es una herramienta útil que facilita el proceso de enseñanza-aprendizaje del software.					X

## 3. COBERTURA Y PROFUNDIDAD

Descripción	Calificación				
	1	2	3	4	5
Los temas abordados son relevantes para el desarrollo de los talleres.					X
El tema presentado se explica de manera detallada.				X	
Los ejemplos expuestos son claros y útiles.					X
El grado de complejidad del taller se ajusta a mis conocimientos.					X

## 4. UTILIDAD

Descripción	Calificación				
	1	2	3	4	5
La interfaz que maneja el software es amigable y facilita el proceso enseñanza-aprendizaje.					X
El taller refuerza el aprendizaje del software.					X
Las imágenes facilitan la explicación de los conceptos y procedimientos.					X
El glosario presentado es útil.				X	
El software utilizado en el taller es una buena herramienta para enseñar Simulación.					X



ENCUESTA DE EVALUACIÓN

Estudiante:	<i>Manuel Cantor Salazar</i>	Código:	<i>2060868</i>
Asignatura:	Técnicas Modernas de Optimización	Código:	23536
Docente:	Ing. Edwin Alberto Garavito Hernández	Fecha:	09 / 04 / 2012
Práctica:	Taller No. 3 – Análisis de Datos con ExpertFit		

Por favor diligencie este formato, teniendo en cuenta la escala de 1 a 5. Marque con una X la calificación correspondiente teniendo en cuenta los siguientes criterios:

1. Deficiente	2. Regular	3. Bueno	4. Muy bueno	5. Excelente
---------------	------------	----------	--------------	--------------

1. PRESENTACION

Descripción	Calificación				
	1	2	3	4	5
La presentación general del material facilita su lectura y utilización como guía de estudio.					X
Las imágenes contenidas en el taller facilitan la comprensión de los conceptos estudiados en el mismo.					X
La proporción, tamaño y distribución del taller es adecuado.				X	
Las instrucciones que se imparten son claras y fáciles de seguir.				X	
El taller esta estructurado de tal manera que contribuye a la comprensión del software.					X



## 2. LENGUAJE

Descripción	Calificación				
	1	2	3	4	5
El taller es explícito y contiene los elementos necesarios para realizar la práctica.				X	
El lenguaje utilizado es claro, comprensible y se ajusta a mis conocimientos.					X
El taller es una herramienta útil que facilita el proceso de enseñanza-aprendizaje del software.					X

## 3. COBERTURA Y PROFUNDIDAD

Descripción	Calificación				
	1	2	3	4	5
Los temas abordados son relevantes para el desarrollo de los talleres.					X
El tema presentado se explica de manera detallada.					X
Los ejemplos expuestos son claros y útiles.					X
El grado de complejidad del taller se ajusta a mis conocimientos.					X

## 4. UTILIDAD

Descripción	Calificación				
	1	2	3	4	5
La interfaz que maneja el software es amigable y facilita el proceso enseñanza-aprendizaje.					X
El taller refuerza el aprendizaje del software.					X
Las imágenes facilitan la explicación de los conceptos y procedimientos.					X
El glosario presentado es útil.				X	
El software utilizado en el taller es una buena herramienta para enseñar Simulación.					X



ENCUESTA DE EVALUACIÓN

Estudiante:	Belkis Yadira Vera Aguilar	Código:	2008904
Asignatura:	Técnicas Modernas de Optimización	Código:	23536
Docente:	Ing. Edwin Alberto Garavito Hernández	Fecha:	09 / 04 / 2012
Práctica:	Taller No. 3 – Análisis de Datos con ExpertFit		

Por favor diligencie este formato, teniendo en cuenta la escala de 1 a 5. Marque con una X la calificación correspondiente teniendo en cuenta los siguientes criterios:

1. Deficiente	2. Regular	3. Bueno	4. Muy bueno	5. Excelente
---------------	------------	----------	--------------	--------------

1. PRESENTACION

Descripción	Calificación				
	1	2	3	4	5
La presentación general del material facilita su lectura y utilización como guía de estudio.					X
Las imágenes contenidas en el taller facilitan la comprensión de los conceptos estudiados en el mismo.					X
La proporción, tamaño y distribución del taller es adecuado.					X
Las instrucciones que se imparten son claras y fáciles de seguir.					X
El taller esta estructurado de tal manera que contribuye a la comprensión del software.					X



## 2. LENGUAJE

Descripción	Calificación				
	1	2	3	4	5
El taller es explícito y contiene los elementos necesarios para realizar la práctica.					X
El lenguaje utilizado es claro, comprensible y se ajusta a mis conocimientos.					X
El taller es una herramienta útil que facilita el proceso de enseñanza-aprendizaje del software.					X

## 3. COBERTURA Y PROFUNDIDAD

Descripción	Calificación				
	1	2	3	4	5
Los temas abordados son relevantes para el desarrollo de los talleres.					X
El tema presentado se explica de manera detallada.					X
Los ejemplos expuestos son claros y útiles.					X
El grado de complejidad del taller se ajusta a mis conocimientos.				X	

## 4. UTILIDAD

Descripción	Calificación				
	1	2	3	4	5
La interfaz que maneja el software es amigable y facilita el proceso enseñanza-aprendizaje.					X
El taller refuerza el aprendizaje del software.					X
Las imágenes facilitan la explicación de los conceptos y procedimientos.					X
El glosario presentado es útil.					X
El software utilizado en el taller es una buena herramienta para enseñar Simulación.					X



ENCUESTA DE EVALUACIÓN

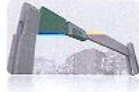
Estudiante:	Leiner Lache Salcedo	Código:	2070233
Asignatura:	Técnicas Modernas de Optimización	Código:	23536
Docente:	Ing. Edwin Alberto Garavito Hernández	Fecha:	09 / 04 / 2012
Práctica:	Taller No. 3 – Análisis de Datos con ExpertFit		

Por favor diligencie este formato, teniendo en cuenta la escala de 1 a 5. Marque con una X la calificación correspondiente teniendo en cuenta los siguientes criterios:

1. Deficiente	2. Regular	3. Bueno	4. Muy bueno	5. Excelente
---------------	------------	----------	--------------	--------------

1. PRESENTACION

Descripción	Calificación				
	1	2	3	4	5
La presentación general del material facilita su lectura y utilización como guía de estudio.					X
Las imágenes contenidas en el taller facilitan la comprensión de los conceptos estudiados en el mismo.					X
La proporción, tamaño y distribución del taller es adecuado.					X
Las instrucciones que se imparten son claras y fáciles de seguir.					X
El taller esta estructurado de tal manera que contribuye a la comprensión del software.					X



## 2. LENGUAJE

Descripción	Calificación				
	1	2	3	4	5
El taller es explícito y contiene los elementos necesarios para realizar la práctica.				X	
El lenguaje utilizado es claro, comprensible y se ajusta a mis conocimientos.					X
El taller es una herramienta útil que facilita el proceso de enseñanza-aprendizaje del software.					X

## 3. COBERTURA Y PROFUNDIDAD

Descripción	Calificación				
	1	2	3	4	5
Los temas abordados son relevantes para el desarrollo de los talleres.					X
El tema presentado se explica de manera detallada.					X
Los ejemplos expuestos son claros y útiles.					X
El grado de complejidad del taller se ajusta a mis conocimientos.					X

## 4. UTILIDAD

Descripción	Calificación				
	1	2	3	4	5
La interfaz que maneja el software es amigable y facilita el proceso enseñanza-aprendizaje.					X
El taller refuerza el aprendizaje del software.					X
Las imágenes facilitan la explicación de los conceptos y procedimientos.					X
El glosario presentado es útil.				X	
El software utilizado en el taller es una buena herramienta para enseñar Simulación.					X



## TALLERES PRÁCTICOS DE SIMULACIÓN CON FLEXSIM

### ENCUESTA DE EVALUACIÓN

<b>Estudiante:</b>	JOHAN ALONSO GUEVARA	<b>Código:</b>	2062533
<b>Asignatura:</b>	Técnicas Modernas de Optimización	<b>Código:</b>	23536
<b>Docente:</b>	Ing. Edwin Alberto Garavito Hernández	<b>Fecha:</b>	09 / 04 / 2012
<b>Práctica:</b>	Taller No. 3 – Análisis de Datos con ExpertFit		

Por favor diligencie este formato, teniendo en cuenta la escala de 1 a 5. Marque con una X la calificación correspondiente teniendo en cuenta los siguientes criterios:

1. Deficiente	2. Regular	3. Bueno	4. Muy bueno	5. Excelente
---------------	------------	----------	--------------	--------------

#### 1. PRESENTACION

Descripción	Calificación				
	1	2	3	4	5
La presentación general del material facilita su lectura y utilización como guía de estudio.					X
Las imágenes contenidas en el taller facilitan la comprensión de los conceptos estudiados en el mismo.					X
La proporción, tamaño y distribución del taller es adecuado.					X
Las instrucciones que se imparten son claras y fáciles de seguir.					X
El taller esta estructurado de tal manera que contribuye a la comprensión del software.					X



## TALLERES PRÁCTICOS DE SIMULACIÓN CON FLEXSIM

### 2. LENGUAJE

Descripción	Calificación				
	1	2	3	4	5
El taller es explícito y contiene los elementos necesarios para realizar la práctica.					X
El lenguaje utilizado es claro, comprensible y se ajusta a mis conocimientos.					X
El taller es una herramienta útil que facilita el proceso de enseñanza-aprendizaje del software.					X

### 3. COBERTURA Y PROFUNDIDAD

Descripción	Calificación				
	1	2	3	4	5
Los temas abordados son relevantes para el desarrollo de los talleres.					X
El tema presentado se explica de manera detallada.					X
Los ejemplos expuestos son claros y útiles.					X
El grado de complejidad del taller se ajusta a mis conocimientos.					X

### 4. UTILIDAD

Descripción	Calificación				
	1	2	3	4	5
La interfaz que maneja el software es amigable y facilita el proceso enseñanza-aprendizaje.					X
El taller refuerza el aprendizaje del software.					X
Las imágenes facilitan la explicación de los conceptos y procedimientos.					X
El glosario presentado es útil.					X
El software utilizado en el taller es una buena herramienta para enseñar Simulación.					X



ENCUESTA DE EVALUACIÓN

Estudiante:	Adriana Marcela Vargas Saavedra	Código:	2071493
Asignatura:	Técnicas Modernas de Optimización	Código:	23536
Docente:	Ing. Edwin Alberto Garavito Hernández	Fecha:	09 / 04 / 2012
Práctica:	Taller No. 3 – Análisis de Datos con ExperFit		

Por favor diligencie este formato, teniendo en cuenta la escala de 1 a 5. Marque con una X la calificación correspondiente teniendo en cuenta los siguientes criterios:

1. Deficiente	2. Regular	3. Bueno	4. Muy bueno	5. Excelente
---------------	------------	----------	--------------	--------------

1. PRESENTACION

Descripción	Calificación				
	1	2	3	4	5
La presentación general del material facilita su lectura y utilización como guía de estudio.					X
Las imágenes contenidas en el taller facilitan la comprensión de los conceptos estudiados en el mismo.					X
La proporción, tamaño y distribución del taller es adecuado.					X
Las instrucciones que se imparten son claras y fáciles de seguir.					X
El taller esta estructurado de tal manera que contribuye a la comprensión del software.					X



## 2. LENGUAJE

Descripción	Calificación				
	1	2	3	4	5
El taller es explícito y contiene los elementos necesarios para realizar la práctica.					X
El lenguaje utilizado es claro, comprensible y se ajusta a mis conocimientos.					X
El taller es una herramienta útil que facilita el proceso de enseñanza-aprendizaje del software.					X

## 3. COBERTURA Y PROFUNDIDAD

Descripción	Calificación				
	1	2	3	4	5
Los temas abordados son relevantes para el desarrollo de los talleres.					X
El tema presentado se explica de manera detallada.					X
Los ejemplos expuestos son claros y útiles.					X
El grado de complejidad del taller se ajusta a mis conocimientos.					X

## 4. UTILIDAD

Descripción	Calificación				
	1	2	3	4	5
La interfaz que maneja el software es amigable y facilita el proceso enseñanza-aprendizaje.					X
El taller refuerza el aprendizaje del software.					X
Las imágenes facilitan la explicación de los conceptos y procedimientos.					X
El glosario presentado es útil.					X
El software utilizado en el taller es una buena herramienta para enseñar Simulación.					X



ENCUESTA DE EVALUACIÓN

Estudiante:	Agced Angel Acosta	Código:	2012211
Docente:	Elwin Gotovito	Fecha:	10/1/2017
Práctica:	Taller No. 3 – Análisis de Datos con Experfit		

Por favor diligencie este formato, teniendo en cuenta la escala de 1 a 5. Marque con una X la calificación correspondiente teniendo en cuenta los siguientes criterios:

1. Deficiente	2. Regular	3. Bueno	4. Muy bueno	5. Excelente
---------------	------------	----------	--------------	--------------

1. PRESENTACION

Descripción	Calificación				
	1	2	3	4	5
La presentación general del material facilita su lectura y utilización como guía de estudio.				X	
Las imágenes contenidas en el taller facilitan la comprensión de los conceptos estudiados en el mismo.				X	
La proporción, tamaño y distribución de las figuras es adecuado.			X		
Las instrucciones que se imparten son claras y fáciles de seguir.					X
El taller esta estructurado de tal manera que contribuye a la comprensión del software.				X	



2. LENGUAJE

Descripción	Calificación				
	1	2	3	4	5
El taller es explícito y contiene los elementos necesarios para realizar la práctica.				X	
El lenguaje utilizado es claro, comprensible y se ajusta a mis conocimientos.				X	
El taller es una herramienta útil que facilita el proceso de enseñanza-aprendizaje del software.			X		

3. COBERTURA Y PROFUNDIDAD

Descripción	Calificación				
	1	2	3	4	5
Los temas abordados son relevantes para el desarrollo de los talleres.				X	
El tema presentado se explica de manera detallada.					X
Los ejemplos expuestos son claros y útiles.				X	
El grado de complejidad del taller se ajusta a mis conocimientos.			X		

4. UTILIDAD

Descripción	Calificación				
	1	2	3	4	5
La interfaz que maneja el software es amigable y facilita el proceso enseñanza-aprendizaje.			X		
El taller refuerza el aprendizaje del software.				X	
Las imágenes facilitan la explicación de los conceptos y procedimientos.				X	
El glosario presentado es útil.				X	
El software utilizado en el taller es una buena herramienta para enseñar Simulación.			X		



ENCUESTA DE EVALUACIÓN

Estudiante:	<i>Carolina Rocio Murillo Maya</i>	Código:	<i>2053151</i>
Docente:	<i>Edwin Garavito</i>	Fecha:	<i>26-04-12</i>
Práctica:	Taller No. 3 – Análisis de Datos con ExperFit		

Por favor diligencie este formato, teniendo en cuenta la escala de 1 a 5. Marque con una X la calificación correspondiente teniendo en cuenta los siguientes criterios:

1. Deficiente	2. Regular	3. Bueno	4. Muy bueno	5. Excelente
---------------	------------	----------	--------------	--------------

1. PRESENTACION

Descripción	Calificación				
	1	2	3	4	5
La presentación general del material facilita su lectura y utilización como guía de estudio.					X
Las imágenes contenidas en el taller facilitan la comprensión de los conceptos estudiados en el mismo.					X
La proporción, tamaño y distribución de las figuras es adecuado.					X
Las instrucciones que se imparten son claras y fáciles de seguir.					X
El taller esta estructurado de tal manera que contribuye a la comprensión del software.					X



2. LENGUAJE

Descripción	Calificación				
	1	2	3	4	5
El taller es explícito y contiene los elementos necesarios para realizar la práctica.					X
El lenguaje utilizado es claro, comprensible y se ajusta a mis conocimientos.					X
El taller es una herramienta útil que facilita el proceso de enseñanza-aprendizaje del software.					X

3. COBERTURA Y PROFUNDIDAD

Descripción	Calificación				
	1	2	3	4	5
Los temas abordados son relevantes para el desarrollo de los talleres.					X
El tema presentado se explica de manera detallada.					X
Los ejemplos expuestos son claros y útiles.					X
El grado de complejidad del taller se ajusta a mis conocimientos.					X

4. UTILIDAD

Descripción	Calificación				
	1	2	3	4	5
La interfaz que maneja el software es amigable y facilita el proceso enseñanza-aprendizaje.					X
El taller refuerza el aprendizaje del software.					X
Las imágenes facilitan la explicación de los conceptos y procedimientos.					X
El glosario presentado es útil.					X
El software utilizado en el taller es una buena herramienta para enseñar Simulación.					X



## TALLERES PRÁCTICOS DE SIMULACIÓN CON FLEXSIM

### ENCUESTA DE EVALUACIÓN

<b>Estudiante:</b>	Karen Martinez Campo	<b>Código:</b>	2062554
<b>Asignatura:</b>	Técnicas Modernas de Optimización	<b>Código:</b>	23536
<b>Docente:</b>	Ing. Edwin Alberto Garavito Hernández	<b>Fecha:</b>	09 / 04 / 2012
<b>Práctica:</b>	Taller No. 3 – Análisis de Datos con ExpertFit		

Por favor diligencie este formato, teniendo en cuenta la escala de 1 a 5. Marque con una X la calificación correspondiente teniendo en cuenta los siguientes criterios:

<b>1. Deficiente</b>	<b>2. Regular</b>	<b>3. Bueno</b>	<b>4. Muy bueno</b>	<b>5. Excelente</b>
----------------------	-------------------	-----------------	---------------------	---------------------

#### 1. PRESENTACION

Descripción	Calificación				
	1	2	3	4	5
La presentación general del material facilita su lectura y utilización como guía de estudio.					X
Las imágenes contenidas en el taller facilitan la comprensión de los conceptos estudiados en el mismo.					X
La proporción, tamaño y distribución del taller es adecuado.					X
Las instrucciones que se imparten son claras y fáciles de seguir.				X	
El taller esta estructurado de tal manera que contribuye a la comprensión del software.					X



## TALLERES PRÁCTICOS DE SIMULACIÓN CON FLEXSIM

### 2. LENGUAJE

Descripción	Calificación				
	1	2	3	4	5
El taller es explícito y contiene los elementos necesarios para realizar la práctica.					x
El lenguaje utilizado es claro, comprensible y se ajusta a mis conocimientos.					x
El taller es una herramienta útil que facilita el proceso de enseñanza-aprendizaje del software.					x

### 3. COBERTURA Y PROFUNDIDAD

Descripción	Calificación				
	1	2	3	4	5
Los temas abordados son relevantes para el desarrollo de los talleres.					x
El tema presentado se explica de manera detallada.					x
Los ejemplos expuestos son claros y útiles.					x
El grado de complejidad del taller se ajusta a mis conocimientos.					x

### 4. UTILIDAD

Descripción	Calificación				
	1	2	3	4	5
La interfaz que maneja el software es amigable y facilita el proceso enseñanza-aprendizaje.					x
El taller refuerza el aprendizaje del software.					x
Las imágenes facilitan la explicación de los conceptos y procedimientos.					x
El glosario presentado es útil.					x
El software utilizado en el taller es una buena herramienta para enseñar Simulación.					x



ENCUESTA DE EVALUACIÓN

Estudiante:	Mónica Janeth Duarte León	Código:	2062805
Docente:	Edwin Garavito	Fecha:	
Práctica:	Taller No. 3 – Análisis de Datos con ExpertFit		

Por favor diligencie este formato, teniendo en cuenta la escala de 1 a 5. Marque con una X la calificación correspondiente teniendo en cuenta los siguientes criterios:

1. Deficiente	2. Regular	3. Bueno	4. Muy bueno	5. Excelente
---------------	------------	----------	--------------	--------------

1. PRESENTACION

Descripción	Calificación				
	1	2	3	4	5
La presentación general del material facilita su lectura y utilización como guía de estudio.					X
Las imágenes contenidas en el taller facilitan la comprensión de los conceptos estudiados en el mismo.					X
La proporción, tamaño y distribución de las figuras es adecuado.					X
Las instrucciones que se imparten son claras y fáciles de seguir.					X
El taller esta estructurado de tal manera que contribuye a la comprensión del software.					X



## 2. LENGUAJE

Descripción	Calificación				
	1	2	3	4	5
El taller es explícito y contiene los elementos necesarios para realizar la práctica.					X
El lenguaje utilizado es claro, comprensible y se ajusta a mis conocimientos.					X
El taller es una herramienta útil que facilita el proceso de enseñanza-aprendizaje del software.					X

## 3. COBERTURA Y PROFUNDIDAD

Descripción	Calificación				
	1	2	3	4	5
Los temas abordados son relevantes para el desarrollo de los talleres.					X
El tema presentado se explica de manera detallada.					X
Los ejemplos expuestos son claros y útiles.					X
El grado de complejidad del taller se ajusta a mis conocimientos.					X

## 4. UTILIDAD

Descripción	Calificación				
	1	2	3	4	5
La interfaz que maneja el software es amigable y facilita el proceso enseñanza-aprendizaje.					X
El taller refuerza el aprendizaje del software.					X
Las imágenes facilitan la explicación de los conceptos y procedimientos.					X
El glosario presentado es útil.					X
El software utilizado en el taller es una buena herramienta para enseñar Simulación.					X



ENCUESTA DE EVALUACIÓN

Estudiante:	Leonardo José Camargo Perca	Código:	206342
Asignatura:	Técnicas Modernas de Optimización	Código:	23536
Docente:	Ing. Edwin Alberto Garavito Hernández	Fecha:	18 / 04 / 2012
Práctica:	Taller No. 4 – Simulación de Escenarios		

Por favor diligencie este formato, teniendo en cuenta la escala de 1 a 5. Marque con una X la calificación correspondiente teniendo en cuenta los siguientes criterios:

1. Deficiente	2. Regular	3. Bueno	4. Muy bueno	5. Excelente
---------------	------------	----------	--------------	--------------

1. PRESENTACION

Descripción	Calificación				
	1	2	3	4	5
La presentación general del material facilita su lectura y utilización como guía de estudio.					X
Las imágenes contenidas en el taller facilitan la comprensión de los conceptos estudiados en el mismo.				X	
La proporción, tamaño y distribución del taller es adecuado.					X
Las instrucciones que se imparten son claras y fáciles de seguir.					X
El taller esta estructurado de tal manera que contribuye a la comprensión del software.					X



## 2. LENGUAJE

Descripción	Calificación				
	1	2	3	4	5
El taller es explícito y contiene los elementos necesarios para realizar la práctica.					X
El lenguaje utilizado es claro, comprensible y se ajusta a mis conocimientos.				X	
El taller es una herramienta útil que facilita el proceso de enseñanza-aprendizaje del software.					X

## 3. COBERTURA Y PROFUNDIDAD

Descripción	Calificación				
	1	2	3	4	5
Los temas abordados son relevantes para el desarrollo de los talleres.					X
El tema presentado se explica de manera detallada.				X	
Los ejemplos expuestos son claros y útiles.					X
El grado de complejidad del taller se ajusta a mis conocimientos.					X

## 4. UTILIDAD

Descripción	Calificación				
	1	2	3	4	5
La interfaz que maneja el software es amigable y facilita el proceso enseñanza-aprendizaje.					X
El taller refuerza el aprendizaje del software.					X
Las imágenes facilitan la explicación de los conceptos y procedimientos.					X
El glosario presentado es útil.				X	
El software utilizado en el taller es una buena herramienta para enseñar Simulación.					X



## TALLERES PRÁCTICOS DE SIMULACIÓN CON FLEXSIM

### ENCUESTA DE EVALUACIÓN

<b>Estudiante:</b>	<i>Manuel Cantor Salazar</i>	<b>Código:</b>	<i>2060868</i>
<b>Asignatura:</b>	Técnicas Modernas de Optimización	<b>Código:</b>	23536
<b>Docente:</b>	Ing. Edwin Alberto Garavito Hernández	<b>Fecha:</b>	18 / 04 / 2012
<b>Práctica:</b>	Taller No. 4 – Simulación de Escenarios		

Por favor diligencie este formato, teniendo en cuenta la escala de 1 a 5. Marque con una X la calificación correspondiente teniendo en cuenta los siguientes criterios:

1. Deficiente	2. Regular	3. Bueno	4. Muy bueno	5. Excelente
---------------	------------	----------	--------------	--------------

#### 1. PRESENTACION

Descripción	Calificación				
	1	2	3	4	5
La presentación general del material facilita su lectura y utilización como guía de estudio.					X
Las imágenes contenidas en el taller facilitan la comprensión de los conceptos estudiados en el mismo.					X
La proporción, tamaño y distribución del taller es adecuado.					X
Las instrucciones que se imparten son claras y fáciles de seguir.					X
El taller esta estructurado de tal manera que contribuye a la comprensión del software.					X



## 2. LENGUAJE

Descripción	Calificación				
	1	2	3	4	5
El taller es explícito y contiene los elementos necesarios para realizar la práctica.					X
El lenguaje utilizado es claro, comprensible y se ajusta a mis conocimientos.				X	
El taller es una herramienta útil que facilita el proceso de enseñanza-aprendizaje del software.					X

## 3. COBERTURA Y PROFUNDIDAD

Descripción	Calificación				
	1	2	3	4	5
Los temas abordados son relevantes para el desarrollo de los talleres.					X
El tema presentado se explica de manera detallada.					X
Los ejemplos expuestos son claros y útiles.					X
El grado de complejidad del taller se ajusta a mis conocimientos.					X

## 4. UTILIDAD

Descripción	Calificación				
	1	2	3	4	5
La interfaz que maneja el software es amigable y facilita el proceso enseñanza-aprendizaje.					X
El taller refuerza el aprendizaje del software.					X
Las imágenes facilitan la explicación de los conceptos y procedimientos.					X
El glosario presentado es útil.				X	
El software utilizado en el taller es una buena herramienta para enseñar Simulación.					X



## TALLERES PRÁCTICOS DE SIMULACIÓN CON FLEXSIM

### ENCUESTA DE EVALUACIÓN

<b>Estudiante:</b>	Belkis Yadira Vera Aguilera	<b>Código:</b>	2060904
<b>Asignatura:</b>	Técnicas Modernas de Optimización	<b>Código:</b>	23536
<b>Docente:</b>	Ing. Edwin Alberto Garavito Hernández	<b>Fecha:</b>	18 / 04 / 2012
<b>Práctica:</b>	Taller No. 4 – Simulación de Escenarios		

Por favor diligencie este formato, teniendo en cuenta la escala de 1 a 5. Marque con una X la calificación correspondiente teniendo en cuenta los siguientes criterios:

1. Deficiente	2. Regular	3. Bueno	4. Muy bueno	5. Excelente
---------------	------------	----------	--------------	--------------

#### 1. PRESENTACION

Descripción	Calificación				
	1	2	3	4	5
La presentación general del material facilita su lectura y utilización como guía de estudio.					X
Las imágenes contenidas en el taller facilitan la comprensión de los conceptos estudiados en el mismo.					X
La proporción, tamaño y distribución del taller es adecuado.					X
Las instrucciones que se imparten son claras y fáciles de seguir.					X
El taller esta estructurado de tal manera que contribuye a la comprensión del software.					X



## 2. LENGUAJE

Descripción	Calificación				
	1	2	3	4	5
El taller es explícito y contiene los elementos necesarios para realizar la práctica.					X
El lenguaje utilizado es claro, comprensible y se ajusta a mis conocimientos.					X
El taller es una herramienta útil que facilita el proceso de enseñanza-aprendizaje del software.					X

## 3. COBERTURA Y PROFUNDIDAD

Descripción	Calificación				
	1	2	3	4	5
Los temas abordados son relevantes para el desarrollo de los talleres.					X
El tema presentado se explica de manera detallada.					X
Los ejemplos expuestos son claros y útiles.					X
El grado de complejidad del taller se ajusta a mis conocimientos.					X

## 4. UTILIDAD

Descripción	Calificación				
	1	2	3	4	5
La interfaz que maneja el software es amigable y facilita el proceso enseñanza-aprendizaje.					X
El taller refuerza el aprendizaje del software.					X
Las imágenes facilitan la explicación de los conceptos y procedimientos.					X
El glosario presentado es útil.					X
El software utilizado en el taller es una buena herramienta para enseñar Simulación.					X



## TALLERES PRÁCTICOS DE SIMULACIÓN CON FLEXSIM

### ENCUESTA DE EVALUACIÓN

Estudiante:	leiner leche salcedo	Código:	2020233
Asignatura:	Técnicas Modernas de Optimización	Código:	23536
Docente:	Ing. Edwin Alberto Garavito Hernández	Fecha:	18 / 04 / 2012
Práctica:	Taller No. 4 – Simulación de Escenarios		

Por favor diligencie este formato, teniendo en cuenta la escala de 1 a 5. Marque con una X la calificación correspondiente teniendo en cuenta los siguientes criterios:

1. Deficiente	2. Regular	3. Bueno	4. Muy bueno	5. Excelente
---------------	------------	----------	--------------	--------------

#### 1. PRESENTACION

Descripción	Calificación				
	1	2	3	4	5
La presentación general del material facilita su lectura y utilización como guía de estudio.					X
Las imágenes contenidas en el taller facilitan la comprensión de los conceptos estudiados en el mismo.					X
La proporción, tamaño y distribución del taller es adecuado.				X	
Las instrucciones que se imparten son claras y fáciles de seguir.					X
El taller esta estructurado de tal manera que contribuye a la comprensión del software.					X



## TALLERES PRÁCTICOS DE SIMULACIÓN CON FLEXSIM

### 2. LENGUAJE

Descripción	Calificación				
	1	2	3	4	5
El taller es explícito y contiene los elementos necesarios para realizar la práctica.					X
El lenguaje utilizado es claro, comprensible y se ajusta a mis conocimientos.					X
El taller es una herramienta útil que facilita el proceso de enseñanza-aprendizaje del software.					X

### 3. COBERTURA Y PROFUNDIDAD

Descripción	Calificación				
	1	2	3	4	5
Los temas abordados son relevantes para el desarrollo de los talleres.					X
El tema presentado se explica de manera detallada.					X
Los ejemplos expuestos son claros y útiles.					X
El grado de complejidad del taller se ajusta a mis conocimientos.				X	

### 4. UTILIDAD

Descripción	Calificación				
	1	2	3	4	5
La interfaz que maneja el software es amigable y facilita el proceso enseñanza-aprendizaje.					X
El taller refuerza el aprendizaje del software.					X
Las imágenes facilitan la explicación de los conceptos y procedimientos.					X
El glosario presentado es útil.				X	
El software utilizado en el taller es una buena herramienta para enseñar Simulación.				X	



## TALLERES PRÁCTICOS DE SIMULACIÓN CON FLEXSIM

### ENCUESTA DE EVALUACIÓN

<b>Estudiante:</b>	JOHAN ALONSO GUEVARA	<b>Código:</b>	2062533
<b>Asignatura:</b>	Técnicas Modernas de Optimización	<b>Código:</b>	23536
<b>Docente:</b>	Ing. Edwin Alberto Garavito Hernández	<b>Fecha:</b>	18 / 04 / 2012
<b>Práctica:</b>	Taller No. 4 – Simulación de Escenarios		

Por favor diligencie este formato, teniendo en cuenta la escala de 1 a 5. Marque con una X la calificación correspondiente teniendo en cuenta los siguientes criterios:

1. Deficiente	2. Regular	3. Bueno	4. Muy bueno	5. Excelente
---------------	------------	----------	--------------	--------------

#### 1. PRESENTACION

Descripción	Calificación				
	1	2	3	4	5
La presentación general del material facilita su lectura y utilización como guía de estudio.					X
Las imágenes contenidas en el taller facilitan la comprensión de los conceptos estudiados en el mismo.					X
La proporción, tamaño y distribución del taller es adecuado.					X
Las instrucciones que se imparten son claras y fáciles de seguir.					X
El taller esta estructurado de tal manera que contribuye a la comprensión del software.					X



## 2. LENGUAJE

Descripción	Calificación				
	1	2	3	4	5
El taller es explícito y contiene los elementos necesarios para realizar la práctica.					X
El lenguaje utilizado es claro, comprensible y se ajusta a mis conocimientos.					X
El taller es una herramienta útil que facilita el proceso de enseñanza-aprendizaje del software.					X

## 3. COBERTURA Y PROFUNDIDAD

Descripción	Calificación				
	1	2	3	4	5
Los temas abordados son relevantes para el desarrollo de los talleres.					X
El tema presentado se explica de manera detallada.					X
Los ejemplos expuestos son claros y útiles.					X
El grado de complejidad del taller se ajusta a mis conocimientos.					X

## 4. UTILIDAD

Descripción	Calificación				
	1	2	3	4	5
La interfaz que maneja el software es amigable y facilita el proceso enseñanza-aprendizaje.					X
El taller refuerza el aprendizaje del software.					X
Las imágenes facilitan la explicación de los conceptos y procedimientos.					X
El glosario presentado es útil.					X
El software utilizado en el taller es una buena herramienta para enseñar Simulación.					X



ENCUESTA DE EVALUACIÓN

Estudiante:	Adriano Marcel Vargas Saavedra	Código:	2091993
Asignatura:	Técnicas Modernas de Optimización	Código:	23536
Docente:	Ing. Edwin Alberto Garavito Hernández	Fecha:	18 / 04 / 2012
Práctica:	Taller No. 4 – Simulación de Escenarios		

Por favor diligencie este formato, teniendo en cuenta la escala de 1 a 5. Marque con una X la calificación correspondiente teniendo en cuenta los siguientes criterios:

1. Deficiente	2. Regular	3. Bueno	4. Muy bueno	5. Excelente
---------------	------------	----------	--------------	--------------

1. PRESENTACION

Descripción	Calificación				
	1	2	3	4	5
La presentación general del material facilita su lectura y utilización como guía de estudio.					X
Las imágenes contenidas en el taller facilitan la comprensión de los conceptos estudiados en el mismo.					X
La proporción, tamaño y distribución del taller es adecuado.					X
Las instrucciones que se imparten son claras y fáciles de seguir.					X
El taller esta estructurado de tal manera que contribuye a la comprensión del software.					X



## 2. LENGUAJE

Descripción	Calificación				
	1	2	3	4	5
El taller es explícito y contiene los elementos necesarios para realizar la práctica.					✓
El lenguaje utilizado es claro, comprensible y se ajusta a mis conocimientos.					✗
El taller es una herramienta útil que facilita el proceso de enseñanza-aprendizaje del software.					✗

## 3. COBERTURA Y PROFUNDIDAD

Descripción	Calificación				
	1	2	3	4	5
Los temas abordados son relevantes para el desarrollo de los talleres.					✗
El tema presentado se explica de manera detallada.					✗
Los ejemplos expuestos son claros y útiles.					✗
El grado de complejidad del taller se ajusta a mis conocimientos.					✗

## 4. UTILIDAD

Descripción	Calificación				
	1	2	3	4	5
La interfaz que maneja el software es amigable y facilita el proceso enseñanza-aprendizaje.					✗
El taller refuerza el aprendizaje del software.					✗
Las imágenes facilitan la explicación de los conceptos y procedimientos.					✗
El glosario presentado es útil.					✗
El software utilizado en el taller es una buena herramienta para enseñar Simulación.					✓



ENCUESTA DE EVALUACIÓN

Estudiante:	Miguel Angel Acared	Código:	2012211
Docente:	Edwin Garavito	Fecha:	
Práctica:	Taller No. 4 – Simulación de Escenarios		

Por favor diligencie este formato, teniendo en cuenta la escala de 1 a 5. Marque con una X la calificación correspondiente teniendo en cuenta los siguientes criterios:

6. Deficiente	7. Regular	8. Bueno	9. Muy bueno	10. Excelente
---------------	------------	----------	--------------	---------------

5. PRESENTACION

Descripción	Calificación				
	1	2	3	4	5
La presentación general del material facilita su lectura y utilización como guía de estudio.					X
Las imágenes contenidas en el taller facilitan la comprensión de los conceptos estudiados en el mismo.				X	
La proporción, tamaño y distribución de las figuras es adecuado.					X
Las instrucciones que se imparten son claras y fáciles de seguir.				X	
El taller esta estructurado de tal manera que contribuye a la comprensión del software.			X		



6. LENGUAJE

Descripción	Calificación				
	1	2	3	4	5
El taller es explícito y contiene los elementos necesarios para realizar la práctica.					X
El lenguaje utilizado es claro, comprensible y se ajusta a mis conocimientos.				X	
El taller es una herramienta útil que facilita el proceso de enseñanza-aprendizaje del software.					X

7. COBERTURA Y PROFUNDIDAD

Descripción	Calificación				
	1	2	3	4	5
Los temas abordados son relevantes para el desarrollo de los talleres.				X	
El tema presentado se explica de manera detallada.			X		
Los ejemplos expuestos son claros y útiles.					X
El grado de complejidad del taller se ajusta a mis conocimientos.				X	

8. UTILIDAD

Descripción	Calificación				
	1	2	3	4	5
La interfaz que maneja el software es amigable y facilita el proceso enseñanza-aprendizaje.					X
El taller refuerza el aprendizaje del software.				X	
Las imágenes facilitan la explicación de los conceptos y procedimientos.					X
El glosario presentado es útil.			X		
El software utilizado en el taller es una buena herramienta para enseñar Simulación.				X	



ENCUESTA DE EVALUACIÓN

Estudiante:	<i>Carolina Rocio Murillo Maya</i>	Código:	<i>2053151</i>
Docente:	<i>Edwin Garavito</i>	Fecha:	<i>26-04-12</i>
Práctica:	Taller No. 4 – Simulación de Escenarios		

Por favor diligencie este formato, teniendo en cuenta la escala de 1 a 5. Marque con una X la calificación correspondiente teniendo en cuenta los siguientes criterios:

1. Deficiente	2. Regular	3. Bueno	4. Muy bueno	5. Excelente
---------------	------------	----------	--------------	--------------

1. PRESENTACION

Descripción	Calificación				
	1	2	3	4	5
La presentación general del material facilita su lectura y utilización como guía de estudio.					X
Las imágenes contenidas en el taller facilitan la comprensión de los conceptos estudiados en el mismo.					X
La proporción, tamaño y distribución de las figuras es adecuado.					X
Las instrucciones que se imparten son claras y fáciles de seguir.					X
El taller esta estructurado de tal manera que contribuye a la comprensión del software.					X



2. LENGUAJE

Descripción	Calificación				
	1	2	3	4	5
El taller es explícito y contiene los elementos necesarios para realizar la práctica.					X
El lenguaje utilizado es claro, comprensible y se ajusta a mis conocimientos.					X
El taller es una herramienta útil que facilita el proceso de enseñanza-aprendizaje del software.					X

3. COBERTURA Y PROFUNDIDAD

Descripción	Calificación				
	1	2	3	4	5
Los temas abordados son relevantes para el desarrollo de los talleres.					X
El tema presentado se explica de manera detallada.					X
Los ejemplos expuestos son claros y útiles.					X
El grado de complejidad del taller se ajusta a mis conocimientos.					X

4. UTILIDAD

Descripción	Calificación				
	1	2	3	4	5
La interfaz que maneja el software es amigable y facilita el proceso enseñanza-aprendizaje.					X
El taller refuerza el aprendizaje del software.					X
Las imágenes facilitan la explicación de los conceptos y procedimientos.					X
El glosario presentado es útil.					X
El software utilizado en el taller es una buena herramienta para enseñar Simulación.					X



## TALLERES PRÁCTICOS DE SIMULACIÓN CON FLEXSIM

### ENCUESTA DE EVALUACIÓN

<b>Estudiante:</b>	Karen Martinez Campo	<b>Código:</b>	2062554
<b>Asignatura:</b>	Técnicas Modernas de Optimización	<b>Código:</b>	23536
<b>Docente:</b>	Ing. Edwin Alberto Garavito Hernández	<b>Fecha:</b>	18 / 04 / 2012
<b>Práctica:</b>	Taller No. 4 – Simulación de Escenarios		

Por favor diligencie este formato, teniendo en cuenta la escala de 1 a 5. Marque con una X la calificación correspondiente teniendo en cuenta los siguientes criterios:

<b>1. Deficiente</b>	<b>2. Regular</b>	<b>3. Bueno</b>	<b>4. Muy bueno</b>	<b>5. Excelente</b>
----------------------	-------------------	-----------------	---------------------	---------------------

#### 1. PRESENTACION

Descripción	Calificación				
	1	2	3	4	5
La presentación general del material facilita su lectura y utilización como guía de estudio.					X
Las imágenes contenidas en el taller facilitan la comprensión de los conceptos estudiados en el mismo.					X
La proporción, tamaño y distribución del taller es adecuado.					X
Las instrucciones que se imparten son claras y fáciles de seguir.				X	
El taller esta estructurado de tal manera que contribuye a la comprensión del software.					X



## TALLERES PRÁCTICOS DE SIMULACIÓN CON FLEXSIM

### 2. LENGUAJE

Descripción	Calificación				
	1	2	3	4	5
El taller es explícito y contiene los elementos necesarios para realizar la práctica.					x
El lenguaje utilizado es claro, comprensible y se ajusta a mis conocimientos.					x
El taller es una herramienta útil que facilita el proceso de enseñanza-aprendizaje del software.					x

### 3. COBERTURA Y PROFUNDIDAD

Descripción	Calificación				
	1	2	3	4	5
Los temas abordados son relevantes para el desarrollo de los talleres.					x
El tema presentado se explica de manera detallada.					x
Los ejemplos expuestos son claros y útiles.					x
El grado de complejidad del taller se ajusta a mis conocimientos.					x

### 4. UTILIDAD

Descripción	Calificación				
	1	2	3	4	5
La interfaz que maneja el software es amigable y facilita el proceso enseñanza-aprendizaje.					x
El taller refuerza el aprendizaje del software.					x
Las imágenes facilitan la explicación de los conceptos y procedimientos.					x
El glosario presentado es útil.					x
El software utilizado en el taller es una buena herramienta para enseñar Simulación.					x



## TALLERES PRÁCTICOS DE SIMULACIÓN CON FLEXSIM

### ENCUESTA DE EVALUACIÓN

Estudiante:	Mónica Janeth Duarte León	Código:	2062805
Docente:	Edwin Garavito	Fecha:	
Práctica:	Taller No. 4 – Simulación de Escenarios		

Por favor diligencie este formato, teniendo en cuenta la escala de 1 a 5. Marque con una X la calificación correspondiente teniendo en cuenta los siguientes criterios:

6. Deficiente	7. Regular	8. Bueno	9. Muy bueno	10. Excelente
---------------	------------	----------	--------------	---------------

#### 5. PRESENTACION

Descripción	Calificación				
	1	2	3	4	5
La presentación general del material facilita su lectura y utilización como guía de estudio.					X
Las imágenes contenidas en el taller facilitan la comprensión de los conceptos estudiados en el mismo.					X
La proporción, tamaño y distribución de las figuras es adecuado.					X
Las instrucciones que se imparten son claras y fáciles de seguir.					X
El taller esta estructurado de tal manera que contribuye a la comprensión del software.					X



6. LENGUAJE

Descripción	Calificación				
	1	2	3	4	5
El taller es explícito y contiene los elementos necesarios para realizar la práctica.					X
El lenguaje utilizado es claro, comprensible y se ajusta a mis conocimientos.					X
El taller es una herramienta útil que facilita el proceso de enseñanza-aprendizaje del software.					X

7. COBERTURA Y PROFUNDIDAD

Descripción	Calificación				
	1	2	3	4	5
Los temas abordados son relevantes para el desarrollo de los talleres.					X
El tema presentado se explica de manera detallada.					X
Los ejemplos expuestos son claros y útiles.					X
El grado de complejidad del taller se ajusta a mis conocimientos.					X

8. UTILIDAD

Descripción	Calificación				
	1	2	3	4	5
La interfaz que maneja el software es amigable y facilita el proceso enseñanza-aprendizaje.					X
El taller refuerza el aprendizaje del software.					X
Las imágenes facilitan la explicación de los conceptos y procedimientos.					X
El glosario presentado es útil.					X
El software utilizado en el taller es una buena herramienta para enseñar Simulación.					X



## TALLERES PRÁCTICOS DE SIMULACIÓN CON FLEXSIM

### ENCUESTA DE EVALUACIÓN

<b>Estudiante:</b>	Leonardo José Camargo Peña	<b>Código:</b>	2063421
<b>Asignatura:</b>	Técnicas Modernas de Optimización	<b>Código:</b>	23536
<b>Docente:</b>	Ing. Edwin Alberto Garavito Hernández	<b>Fecha:</b>	26 / 04 / 2012
<b>Práctica:</b>	Taller No. 5 – Sistemas de Control de Producción		

Por favor diligencie este formato, teniendo en cuenta la escala de 1 a 5. Marque con una X la calificación correspondiente teniendo en cuenta los siguientes criterios:

1. Deficiente	2. Regular	3. Bueno	4. Muy bueno	5. Excelente
---------------	------------	----------	--------------	--------------

#### 1. PRESENTACION

Descripción	Calificación				
	1	2	3	4	5
La presentación general del material facilita su lectura y utilización como guía de estudio.					X
Las imágenes contenidas en el taller facilitan la comprensión de los conceptos estudiados en el mismo.					X
La proporción, tamaño y distribución del taller es adecuado.					X
Las instrucciones que se imparten son claras y fáciles de seguir.					X
El taller esta estructurado de tal manera que contribuye a la comprensión del software.					X



## 2. LENGUAJE

Descripción	Calificación				
	1	2	3	4	5
El taller es explícito y contiene los elementos necesarios para realizar la práctica.					X
El lenguaje utilizado es claro, comprensible y se ajusta a mis conocimientos.					X
El taller es una herramienta útil que facilita el proceso de enseñanza-aprendizaje del software.					X

## 3. COBERTURA Y PROFUNDIDAD

Descripción	Calificación				
	1	2	3	4	5
Los temas abordados son relevantes para el desarrollo de los talleres.					X
El tema presentado se explica de manera detallada.					X
Los ejemplos expuestos son claros y útiles.				X	
El grado de complejidad del taller se ajusta a mis conocimientos.					X

## 4. UTILIDAD

Descripción	Calificación				
	1	2	3	4	5
La interfaz que maneja el software es amigable y facilita el proceso enseñanza-aprendizaje.					X
El taller refuerza el aprendizaje del software.					X
Las imágenes facilitan la explicación de los conceptos y procedimientos.				X	
El glosario presentado es útil.					X
El software utilizado en el taller es una buena herramienta para enseñar Simulación.					X



ENCUESTA DE EVALUACIÓN

Estudiante:	<i>Manuel Cantor Salazar</i>	Código:	<i>2060868</i>
Asignatura:	Técnicas Modernas de Optimización	Código:	23536
Docente:	Ing. Edwin Alberto Garavito Hernández	Fecha:	26 / 04 / 2012
Práctica:	Taller No. 5 – Sistemas de Control de Producción		

Por favor diligencie este formato, teniendo en cuenta la escala de 1 a 5. Marque con una X la calificación correspondiente teniendo en cuenta los siguientes criterios:

1. Deficiente	2. Regular	3. Bueno	4. Muy bueno	5. Excelente
---------------	------------	----------	--------------	--------------

1. PRESENTACION

Descripción	Calificación				
	1	2	3	4	5
La presentación general del material facilita su lectura y utilización como guía de estudio.					X
Las imágenes contenidas en el taller facilitan la comprensión de los conceptos estudiados en el mismo.				X	
La proporción, tamaño y distribución del taller es adecuado.					X
Las instrucciones que se imparten son claras y fáciles de seguir.					X
El taller esta estructurado de tal manera que contribuye a la comprensión del software.				X	



### TALLERES PRÁCTICOS DE SIMULACIÓN CON FLEXSIM

Descripción	Calificación				
	1	2	3	4	5
El taller es explícito y contiene los elementos necesarios para realizar la práctica.					X
El lenguaje utilizado es claro, comprensible y se ajusta a mis conocimientos.					X
El taller es una herramienta útil que facilita el proceso de enseñanza-aprendizaje del software.					X

### 3. COBERTURA Y PROFUNDIDAD

Descripción	Calificación				
	1	2	3	4	5
Los temas abordados son relevantes para el desarrollo de los talleres.					X
El tema presentado se explica de manera detallada.					X
Los ejemplos expuestos son claros y útiles.					X
El grado de complejidad del taller se ajusta a mis conocimientos.					X

### 4. UTILIDAD

Descripción	Calificación				
	1	2	3	4	5
La interfaz que maneja el software es amigable y facilita el proceso enseñanza-aprendizaje.					X
El taller refuerza el aprendizaje del software.					X
Las imágenes facilitan la explicación de los conceptos y procedimientos.					X
El glosario presentado es útil.				X	
El software utilizado en el taller es una buena herramienta para enseñar Simulación.					X



## TALLERES PRÁCTICOS DE SIMULACIÓN CON FLEXSIM

### ENCUESTA DE EVALUACIÓN

<b>Estudiante:</b>	Belkis Yadira Vera Aguilar	<b>Código:</b>	20509104
<b>Asignatura:</b>	Técnicas Modernas de Optimización	<b>Código:</b>	23536
<b>Docente:</b>	Ing. Edwin Alberto Garavito Hernández	<b>Fecha:</b>	26 / 04 / 2012
<b>Práctica:</b>	Taller No. 5 – Sistemas de Control de Producción		

Por favor diligencie este formato, teniendo en cuenta la escala de 1 a 5. Marque con una X la calificación correspondiente teniendo en cuenta los siguientes criterios:

1. Deficiente	2. Regular	3. Bueno	4. Muy bueno	5. Excelente
---------------	------------	----------	--------------	--------------

#### 1. PRESENTACION

Descripción	Calificación				
	1	2	3	4	5
La presentación general del material facilita su lectura y utilización como guía de estudio.					X
Las imágenes contenidas en el taller facilitan la comprensión de los conceptos estudiados en el mismo.				X	
La proporción, tamaño y distribución del taller es adecuado.					X
Las instrucciones que se imparten son claras y fáciles de seguir.					X
El taller esta estructurado de tal manera que contribuye a la comprensión del software.					X



## 2. LENGUAJE

Descripción	Calificación				
	1	2	3	4	5
El taller es explícito y contiene los elementos necesarios para realizar la práctica.					X
El lenguaje utilizado es claro, comprensible y se ajusta a mis conocimientos.					X
El taller es una herramienta útil que facilita el proceso de enseñanza-aprendizaje del software.					X

## 3. COBERTURA Y PROFUNDIDAD

Descripción	Calificación				
	1	2	3	4	5
Los temas abordados son relevantes para el desarrollo de los talleres.					X
El tema presentado se explica de manera detallada.					X
Los ejemplos expuestos son claros y útiles.					X
El grado de complejidad del taller se ajusta a mis conocimientos.					X

## 4. UTILIDAD

Descripción	Calificación				
	1	2	3	4	5
La interfaz que maneja el software es amigable y facilita el proceso enseñanza-aprendizaje.					X
El taller refuerza el aprendizaje del software.					X
Las imágenes facilitan la explicación de los conceptos y procedimientos.					X
El glosario presentado es útil.					X
El software utilizado en el taller es una buena herramienta para enseñar Simulación.					X



TALLERES PRÁCTICOS DE SIMULACIÓN CON FLEXSIM

ENCUESTA DE EVALUACIÓN

Estudiante:	<i>Leine Lache Salcedo</i>	Código:	<i>2070233</i>
Asignatura:	Técnicas Modernas de Optimización	Código:	23536
Docente:	Ing. Edwin Alberto Garavito Hernández	Fecha:	26 / 04 / 2012
Práctica:	Taller No. 5 – Sistemas de Control de Producción		

Por favor diligencie este formato, teniendo en cuenta la escala de 1 a 5. Marque con una X la calificación correspondiente teniendo en cuenta los siguientes criterios:

1. Deficiente	2. Regular	3. Bueno	4. Muy bueno	5. Excelente
---------------	------------	----------	--------------	--------------

1. PRESENTACION

Descripción	Calificación				
	1	2	3	4	5
La presentación general del material facilita su lectura y utilización como guía de estudio.					X
Las imágenes contenidas en el taller facilitan la comprensión de los conceptos estudiados en el mismo.					X
La proporción, tamaño y distribución del taller es adecuado.				X	
Las instrucciones que se imparten son claras y fáciles de seguir.					X
El taller esta estructurado de tal manera que contribuye a la comprensión del software.					X



## 2. LENGUAJE

Descripción	Calificación				
	1	2	3	4	5
El taller es explícito y contiene los elementos necesarios para realizar la práctica.					X
El lenguaje utilizado es claro, comprensible y se ajusta a mis conocimientos.					X
El taller es una herramienta útil que facilita el proceso de enseñanza-aprendizaje del software.					X

## 3. COBERTURA Y PROFUNDIDAD

Descripción	Calificación				
	1	2	3	4	5
Los temas abordados son relevantes para el desarrollo de los talleres.					X
El tema presentado se explica de manera detallada.					X
Los ejemplos expuestos son claros y útiles.					X
El grado de complejidad del taller se ajusta a mis conocimientos.				X	

## 4. UTILIDAD

Descripción	Calificación				
	1	2	3	4	5
La interfaz que maneja el software es amigable y facilita el proceso enseñanza-aprendizaje.					X
El taller refuerza el aprendizaje del software.					X
Las imágenes facilitan la explicación de los conceptos y procedimientos.					X
El glosario presentado es útil.				X	
El software utilizado en el taller es una buena herramienta para enseñar Simulación.					X



## TALLERES PRÁCTICOS DE SIMULACIÓN CON FLEXSIM

### ENCUESTA DE EVALUACIÓN

<b>Estudiante:</b>	JOHAN ALONSO GUEVARA	<b>Código:</b>	2062533
<b>Asignatura:</b>	Técnicas Modernas de Optimización	<b>Código:</b>	23536
<b>Docente:</b>	Ing. Edwin Alberto Garavito Hernández	<b>Fecha:</b>	26 / 04 / 2012
<b>Práctica:</b>	Taller No. 5 – Sistemas de Control de Producción		

Por favor diligencie este formato, teniendo en cuenta la escala de 1 a 5. Marque con una X la calificación correspondiente teniendo en cuenta los siguientes criterios:

1. Deficiente	2. Regular	3. Bueno	4. Muy bueno	5. Excelente
---------------	------------	----------	--------------	--------------

#### 1. PRESENTACION

Descripción	Calificación				
	1	2	3	4	5
La presentación general del material facilita su lectura y utilización como guía de estudio.					X
Las imágenes contenidas en el taller facilitan la comprensión de los conceptos estudiados en el mismo.					X
La proporción, tamaño y distribución del taller es adecuado.					X
Las instrucciones que se imparten son claras y fáciles de seguir.					X
El taller esta estructurado de tal manera que contribuye a la comprensión del software.					X



## 2. LENGUAJE

Descripción	Calificación				
	1	2	3	4	5
El taller es explícito y contiene los elementos necesarios para realizar la práctica.					X
El lenguaje utilizado es claro, comprensible y se ajusta a mis conocimientos.					X
El taller es una herramienta útil que facilita el proceso de enseñanza-aprendizaje del software.					X

## 3. COBERTURA Y PROFUNDIDAD

Descripción	Calificación				
	1	2	3	4	5
Los temas abordados son relevantes para el desarrollo de los talleres.					X
El tema presentado se explica de manera detallada.					X
Los ejemplos expuestos son claros y útiles.					X
El grado de complejidad del taller se ajusta a mis conocimientos.					X

## 4. UTILIDAD

Descripción	Calificación				
	1	2	3	4	5
La interfaz que maneja el software es amigable y facilita el proceso enseñanza-aprendizaje.					X
El taller refuerza el aprendizaje del software.					X
Las imágenes facilitan la explicación de los conceptos y procedimientos.					X
El glosario presentado es útil.					X
El software utilizado en el taller es una buena herramienta para enseñar Simulación.					X



## TALLERES PRÁCTICOS DE SIMULACIÓN CON FLEXSIM

### ENCUESTA DE EVALUACIÓN

<b>Estudiante:</b>	Adriano Marcelo Vargas Saucedo	<b>Código:</b>	2071998
<b>Asignatura:</b>	Técnicas Modernas de Optimización	<b>Código:</b>	23536
<b>Docente:</b>	Ing. Edwin Alberto Garavito Hernández	<b>Fecha:</b>	26 / 04 / 2012
<b>Práctica:</b>	Taller No. 5 – Sistemas de Control de Producción		

Por favor diligencie este formato, teniendo en cuenta la escala de 1 a 5. Marque con una X la calificación correspondiente teniendo en cuenta los siguientes criterios:

1. Deficiente	2. Regular	3. Bueno	4. Muy bueno	5. Excelente
---------------	------------	----------	--------------	--------------

#### 1. PRESENTACION

Descripción	Calificación				
	1	2	3	4	5
La presentación general del material facilita su lectura y utilización como guía de estudio.					X
Las imágenes contenidas en el taller facilitan la comprensión de los conceptos estudiados en el mismo.					X
La proporción, tamaño y distribución del taller es adecuado.					X
Las instrucciones que se imparten son claras y fáciles de seguir.					X
El taller esta estructurado de tal manera que contribuye a la comprensión del software.					X



## TALLERES PRÁCTICOS DE SIMULACIÓN CON FLEXSIM

### 2. LENGUAJE

Descripción	Calificación				
	1	2	3	4	5
El taller es explícito y contiene los elementos necesarios para realizar la práctica.					X
El lenguaje utilizado es claro, comprensible y se ajusta a mis conocimientos.					X
El taller es una herramienta útil que facilita el proceso de enseñanza-aprendizaje del software.					X

### 3. COBERTURA Y PROFUNDIDAD

Descripción	Calificación				
	1	2	3	4	5
Los temas abordados son relevantes para el desarrollo de los talleres.					X
El tema presentado se explica de manera detallada.					X
Los ejemplos expuestos son claros y útiles.					X
El grado de complejidad del taller se ajusta a mis conocimientos.					X

### 4. UTILIDAD

Descripción	Calificación				
	1	2	3	4	5
La interfaz que maneja el software es amigable y facilita el proceso enseñanza-aprendizaje.					X
El taller refuerza el aprendizaje del software.					X
Las imágenes facilitan la explicación de los conceptos y procedimientos.					X
El glosario presentado es útil.					X
El software utilizado en el taller es una buena herramienta para enseñar Simulación.					X



ENCUESTA DE EVALUACIÓN

Estudiante:	Miguel Angel Acosta	Código:	2012211
Docente:	E. Javier Garavito	Fecha:	
Práctica:	Taller No. 5 - Sistemas de Control de Producción		

Por favor diligencie este formato, teniendo en cuenta la escala de 1 a 5. Marque con una X la calificación correspondiente teniendo en cuenta los siguientes criterios:

11. Deficiente	12. Regular	13. Bueno	14. Muy bueno	15. Excelente
----------------	-------------	-----------	---------------	---------------

9. PRESENTACION

Descripción	Calificación				
	1	2	3	4	5
La presentación general del material facilita su lectura y utilización como guía de estudio.				X	
Las imágenes contenidas en el taller facilitan la comprensión de los conceptos estudiados en el mismo.					X
La proporción, tamaño y distribución de las figuras es adecuado.				X	
Las instrucciones que se imparten son claras y fáciles de seguir.			X		
El taller esta estructurado de tal manera que contribuye a la comprensión del software.					X



10. LENGUAJE

Descripción	Calificación				
	1	2	3	4	5
El taller es explícito y contiene los elementos necesarios para realizar la práctica.			X		
El lenguaje utilizado es claro, comprensible y se ajusta a mis conocimientos.				X	
El taller es una herramienta útil que facilita el proceso de enseñanza-aprendizaje del software.					X

11. COBERTURA Y PROFUNDIDAD

Descripción	Calificación				
	1	2	3	4	5
Los temas abordados son relevantes para el desarrollo de los talleres.				X	
El tema presentado se explica de manera detallada.					X
Los ejemplos expuestos son claros y útiles.				X	
El grado de complejidad del taller se ajusta a mis conocimientos.			X		

12. UTILIDAD

Descripción	Calificación				
	1	2	3	4	5
La interfaz que maneja el software es amigable y facilita el proceso enseñanza-aprendizaje.				X	
El taller refuerza el aprendizaje del software.				X	
Las imágenes facilitan la explicación de los conceptos y procedimientos.					X
El glosario presentado es útil.			X		
El software utilizado en el taller es una buena herramienta para enseñar Simulación.				X	



ENCUESTA DE EVALUACIÓN

Estudiante:	Carlos René Morillo Morán	Código:	2053151
Docente:	Edwin Garavito	Fecha:	26-04-12
Práctica:	Taller No. 5 – Sistemas de Control de Producción		

Por favor diligencie este formato, teniendo en cuenta la escala de 1 a 5. Marque con una X la calificación correspondiente teniendo en cuenta los siguientes criterios:

11. Deficiente	12. Regular	13. Bueno	14. Muy bueno	15. Excelente
----------------	-------------	-----------	---------------	---------------

9. PRESENTACION

Descripción	Calificación				
	1	2	3	4	5
La presentación general del material facilita su lectura y utilización como guía de estudio.					X
Las imágenes contenidas en el taller facilitan la comprensión de los conceptos estudiados en el mismo.					X
La proporción, tamaño y distribución de las figuras es adecuado.					X
Las instrucciones que se imparten son claras y fáciles de seguir.					X
El taller esta estructurado de tal manera que contribuye a la comprensión del software.					X



## TALLERES PRÁCTICOS DE SIMULACIÓN CON FLEXSIM

### 10. LENGUAJE

Descripción	Calificación				
	1	2	3	4	5
El taller es explícito y contiene los elementos necesarios para realizar la práctica.					X
El lenguaje utilizado es claro, comprensible y se ajusta a mis conocimientos.					X
El taller es una herramienta útil que facilita el proceso de enseñanza-aprendizaje del software.					X

### 11. COBERTURA Y PROFUNDIDAD

Descripción	Calificación				
	1	2	3	4	5
Los temas abordados son relevantes para el desarrollo de los talleres.					X
El tema presentado se explica de manera detallada.					X
Los ejemplos expuestos son claros y útiles.					X
El grado de complejidad del taller se ajusta a mis conocimientos.					X

### 12. UTILIDAD

Descripción	Calificación				
	1	2	3	4	5
La interfaz que maneja el software es amigable y facilita el proceso enseñanza-aprendizaje.					X
El taller refuerza el aprendizaje del software.					X
Las imágenes facilitan la explicación de los conceptos y procedimientos.					X
El glosario presentado es útil.					X
El software utilizado en el taller es una buena herramienta para enseñar Simulación.					X



## TALLERES PRÁCTICOS DE SIMULACIÓN CON FLEXSIM

### ENCUESTA DE EVALUACIÓN

<b>Estudiante:</b>	Karen Martínez Campo	<b>Código:</b>	2062554
<b>Asignatura:</b>	Técnicas Modernas de Optimización	<b>Código:</b>	23536
<b>Docente:</b>	Ing. Edwin Alberto Garavito Hernández	<b>Fecha:</b>	26 / 04 / 2012
<b>Práctica:</b>	<b>Taller No. 5 – Sistemas de Control de Producción</b>		

Por favor diligencie este formato, teniendo en cuenta la escala de 1 a 5. Marque con una X la calificación correspondiente teniendo en cuenta los siguientes criterios:

<b>1. Deficiente</b>	<b>2. Regular</b>	<b>3. Bueno</b>	<b>4. Muy bueno</b>	<b>5. Excelente</b>
----------------------	-------------------	-----------------	---------------------	---------------------

#### 1. PRESENTACION

Descripción	Calificación				
	1	2	3	4	5
La presentación general del material facilita su lectura y utilización como guía de estudio.					X
Las imágenes contenidas en el taller facilitan la comprensión de los conceptos estudiados en el mismo.					X
La proporción, tamaño y distribución del taller es adecuado.					X
Las instrucciones que se imparten son claras y fáciles de seguir.					X
El taller esta estructurado de tal manera que contribuye a la comprensión del software.					X



## TALLERES PRÁCTICOS DE SIMULACIÓN CON FLEXSIM

### 2. LENGUAJE

Descripción	Calificación				
	1	2	3	4	5
El taller es explícito y contiene los elementos necesarios para realizar la práctica.					✓
El lenguaje utilizado es claro, comprensible y se ajusta a mis conocimientos.					✓
El taller es una herramienta útil que facilita el proceso de enseñanza-aprendizaje del software.					✓

### 3. COBERTURA Y PROFUNDIDAD

Descripción	Calificación				
	1	2	3	4	5
Los temas abordados son relevantes para el desarrollo de los talleres.					✗
El tema presentado se explica de manera detallada.					✗
Los ejemplos expuestos son claros y útiles.					✓
El grado de complejidad del taller se ajusta a mis conocimientos.					✓

### 4. UTILIDAD

Descripción	Calificación				
	1	2	3	4	5
La interfaz que maneja el software es amigable y facilita el proceso enseñanza-aprendizaje.					✓
El taller refuerza el aprendizaje del software.					✗
Las imágenes facilitan la explicación de los conceptos y procedimientos.					✗
El glosario presentado es útil.					✗
El software utilizado en el taller es una buena herramienta para enseñar Simulación.					✓



ENCUESTA DE EVALUACIÓN

Estudiante:	Mónica Janeth Duarte León	Código:	2062805
Docente:	Edwin Garavito	Fecha:	
Práctica:	Taller No. 5 – Sistemas de Control de Producción		

Por favor diligencie este formato, teniendo en cuenta la escala de 1 a 5. Marque con una X la calificación correspondiente teniendo en cuenta los siguientes criterios:

6. Deficiente	7. Regular	8. Bueno	9. Muy bueno	10. Excelente
---------------	------------	----------	--------------	---------------

5. PRESENTACION

Descripción	Calificación				
	1	2	3	4	5
La presentación general del material facilita su lectura y utilización como guía de estudio.					X
Las imágenes contenidas en el taller facilitan la comprensión de los conceptos estudiados en el mismo.					X
La proporción, tamaño y distribución de las figuras es adecuado.					X
Las instrucciones que se imparten son claras y fáciles de seguir.					X
El taller esta estructurado de tal manera que contribuye a la comprensión del software.					X



6. LENGUAJE

Descripción	Calificación				
	1	2	3	4	5
El taller es explícito y contiene los elementos necesarios para realizar la práctica.					X
El lenguaje utilizado es claro, comprensible y se ajusta a mis conocimientos.					X
El taller es una herramienta útil que facilita el proceso de enseñanza-aprendizaje del software.					X

LISTA DE EVALUACIÓN

7. COBERTURA Y PROFUNDIDAD

Descripción	Calificación				
	1	2	3	4	5
Los temas abordados son relevantes para el desarrollo de los talleres.					X
El tema presentado se explica de manera detallada.					X
Los ejemplos expuestos son claros y útiles.					X
El grado de complejidad del taller se ajusta a mis conocimientos.					X

8. UTILIDAD

Descripción	Calificación				
	1	2	3	4	5
La interfaz que maneja el software es amigable y facilita el proceso enseñanza-aprendizaje.					X
El taller refuerza el aprendizaje del software.					X
Las imágenes facilitan la explicación de los conceptos y procedimientos.					X
El glosario presentado es útil.					X
El software utilizado en el taller es una buena herramienta para enseñar Simulación.					X