

# Flujo en redes con Simio

Módulo 5 (*with slides in English*)

Angel A. Juan

PID\_00209187

*Ninguna parte de esta publicación, incluido el diseño general y la cubierta, puede ser copiada, reproducida, almacenada o transmitida de ninguna forma, ni por ningún medio, sea éste eléctrico, químico, mecánico, óptico, grabación, fotocopia, o cualquier otro, sin la previa autorización escrita de los titulares del copyright.*

# Índice

<b>Introducción</b> .....	5
<b>Objetivos</b> .....	6
<b>1. Centro de mantenimiento e inspección de aeronaves</b> .....	7
1.1. Descripción del sistema .....	7
1.2. Creación del modelo – Objetos básicos .....	7
1.3. Creación del modelo – Símbolos .....	8
1.4. Ejecución del modelo .....	9
<b>2. Modelado de colas balanceadas</b> .....	10
2.1. Descripción del sistema .....	10
2.2. Creación del modelo – Objetos básicos .....	10
2.3. Creación del modelo – Lista de destinos .....	11
2.4. Ejecución del modelo .....	12
<b>3. Modelado de tareas secuenciales (<i>Job-Shops</i>)</b> .....	13
3.1. Descripción del sistema .....	13
3.2. Creación del modelo – Objetos básicos .....	13
3.3. Creación del modelo – Tablas de secuencia .....	14
3.4. Ejecución del modelo .....	15
<b>Actividades</b> .....	17
<b>Bibliografía</b> .....	17



## Introducción

En el presente módulo se introducen diferentes ejemplos relativos al modelado y simulación de flujo en redes. El primer ejemplo presenta un proceso básico de dos etapas con retroalimentación, de modo que algunas de las entidades son devueltas de la segunda a la primera etapa al no superar algún control de calidad. El segundo ejemplo ilustra cómo balancear diferentes colas de modo que las entidades se dirijan siempre hacia la línea de espera de menor tamaño. El tercer ejemplo ilustra un proceso sencillo de *job-shop*, en el que entidades de distinto tipo siguen secuencias de procesado diferentes a través de la cadena de producción. El módulo incluye también una relación de referencias y enlaces complementarios, así como con una propuesta de actividades.

### Observation

Ask your Instructor for the slides associated with this module.

## Objetivos

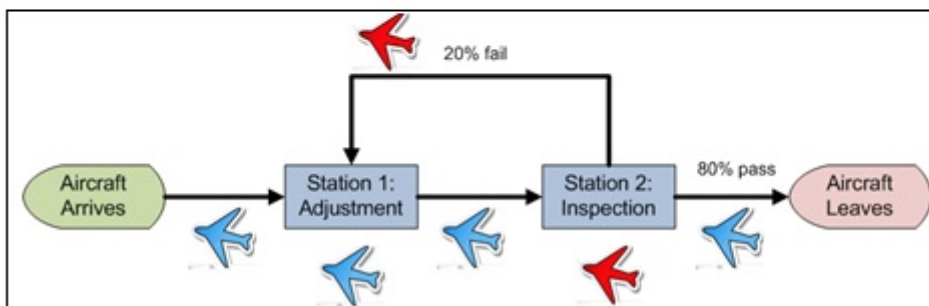
1. Adquirir competencias básicas para modelar y simular flujos de entidades en sistemas de red.
2. Aprender algunas de las opciones que ofrece Simio para el modelado y simulación de flujos en redes.
3. Descubrir y explorar potenciales ejemplos de aplicación de los conceptos y habilidades adquiridos.

## 1. Centro de mantenimiento e inspección de aeronaves

### 1.1. Descripción del sistema

El primer sistema que consideramos es un centro de mantenimiento e inspección de aeronaves. Las aeronaves llegan a una estación de mantenimiento, en la que son revisadas y reparadas por un primer equipo de ingenieros (figura 1). Después, las aeronaves son inspeccionadas en una segunda estación por otro equipo de ingenieros. Alrededor del 20 % de las aeronaves no superan esta inspección y son reenviadas de vuelta a la primera estación para una nueva revisión. La aeronave rechazada en la fase de inspección es representada por un símbolo rojo, el cual mantiene hasta superar con éxito la fase de inspección.

Figura 1. Representación gráfica del sistema

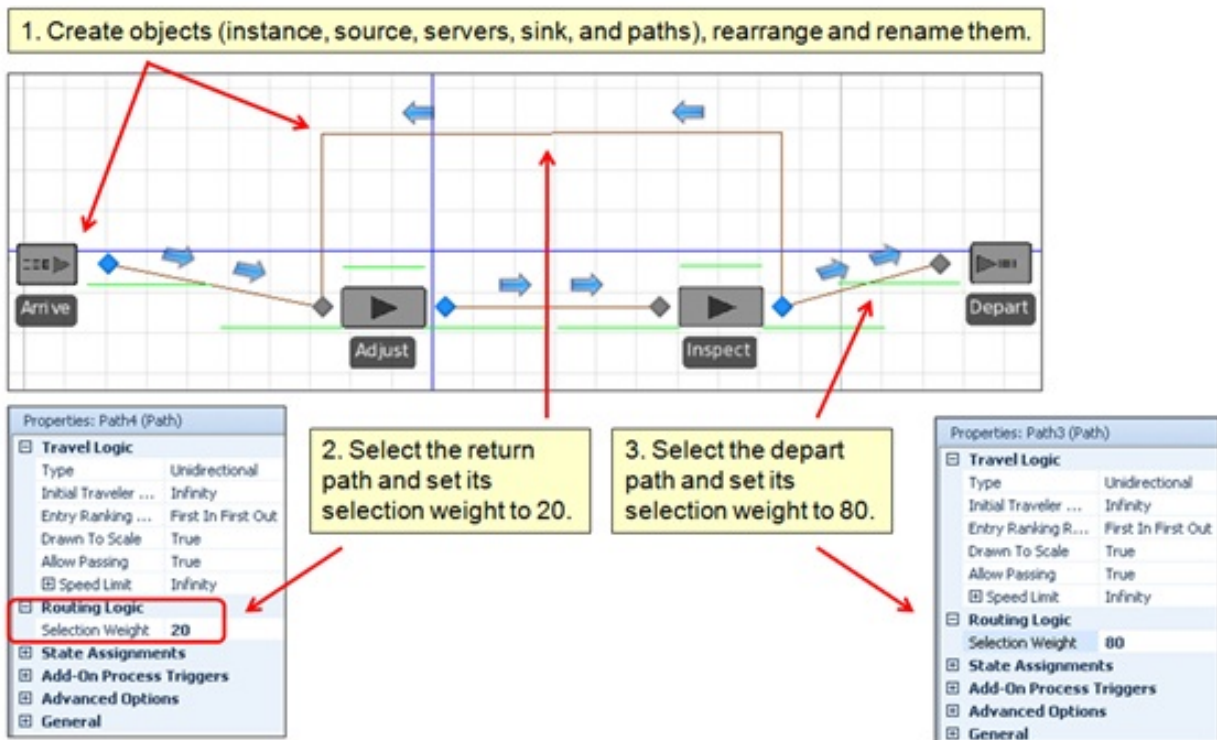


### 1.2. Creación del modelo – Objetos básicos

Los pasos para crear los objetos básicos del modelo son los siguientes (figura 2):

- 1) Crea, ordena, y renombra los objetos instancia, fuente, servidores (estaciones), sumidero, y caminos.
- 2) Selecciona el camino de retorno y fija su porcentaje de selección a 20.
- 3) Selecciona el camino de salida y fija su porcentaje de selección a 80.

Figura 2. Creación del modelo – Objetos básicos



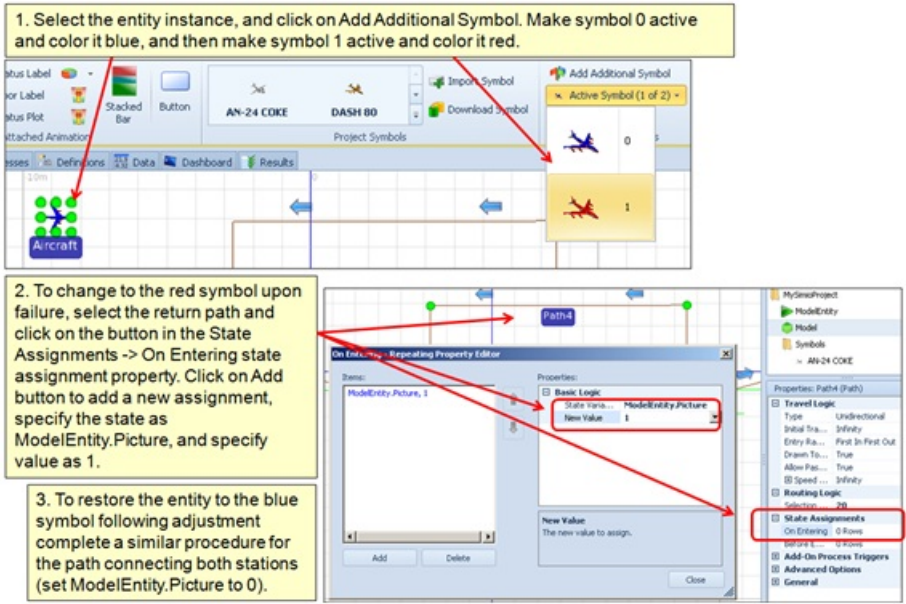
### 1.3. Creación del modelo – Símbolos

A fin de añadir los símbolos de colores (aeronave azul por defecto y aeronave roja cuando el aparato no ha sido validado durante la fase de inspección), se procede como sigue (figura 3):

- 1) Selecciona la entidad y usa el menú de añadir símbolo para: (a) activar el símbolo 0 y asignarle el color azul; y (b) activar, posteriormente, el símbolo 1 y asignarle el color rojo.
- 2) Para cambiar a rojo el color del símbolo cuando la aeronave no resulte validada en la fase de inspección, selecciona: (a) el camino de regreso; (b) la propiedad *StateAssignments* → *OnEnteringstateassignmentproperty*; y (c) la opción de añadir una nueva asignación, especificando el estado como *ModelEntity.Picture* con valor 1.
- 3) Para restablecer la entidad al símbolo de color azul una vez reparada la aeronave, completa un procedimiento similar al anterior para el camino que conecta ambas estaciones, especificando el estado como *ModelEntity.Picture* con valor 0.



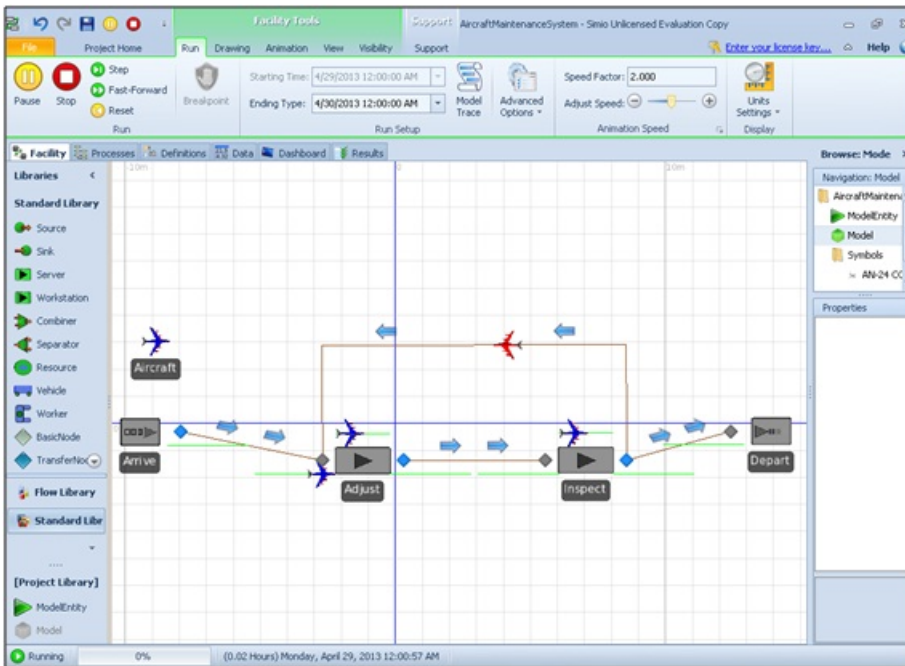
Figura 3. Creación del modelo – Símbolos



### 1.4. Ejecución del modelo

Una vez finalizada la fase de modelado, se procede a iniciar la simulación del mismo. El resultado debería ser similar al que se muestra en la figura 4. Observa cómo aproximadamente un 20 % de las aeronaves cambian de color y son devueltas a la estación de mantenimiento cuando son rechazadas durante la fase de inspección. Por el contrario, aproximadamente un 80 % de las aeronaves pasa con éxito la fase de inspección y se dirige hacia la salida o sumidero.

Figura 4. Ejecución del modelo

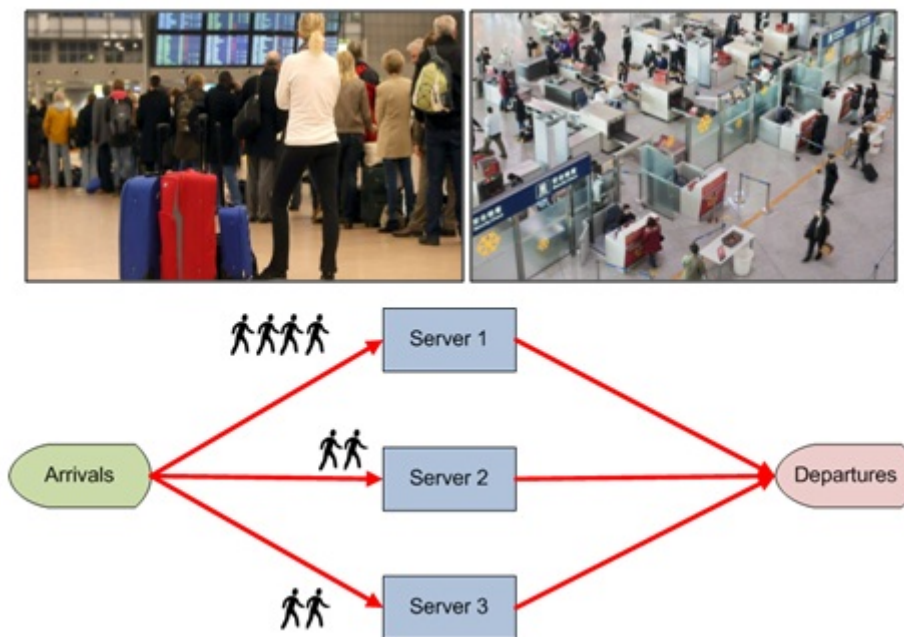


## 2. Modelado de colas balanceadas

### 2.1. Descripción del sistema

En sistemas de gestión de colas con varios servidores y colas (figura 5), las entidades que llegan al sistema tienden a unirse a la cola que tiene un menor número de entidades en espera. Considera un sistema con tres servidores: el servidor 1 tiene un tiempo de procesado constante de 20 minutos; el servidor 2 tiene un tiempo de procesado aleatorio que sigue una distribución  $Triang(1, 3, 8)$  (en minutos); y el servidor 3 tiene un tiempo de procesado constante de 0.1 minutos.

Figura 5. Representación gráfica del sistema



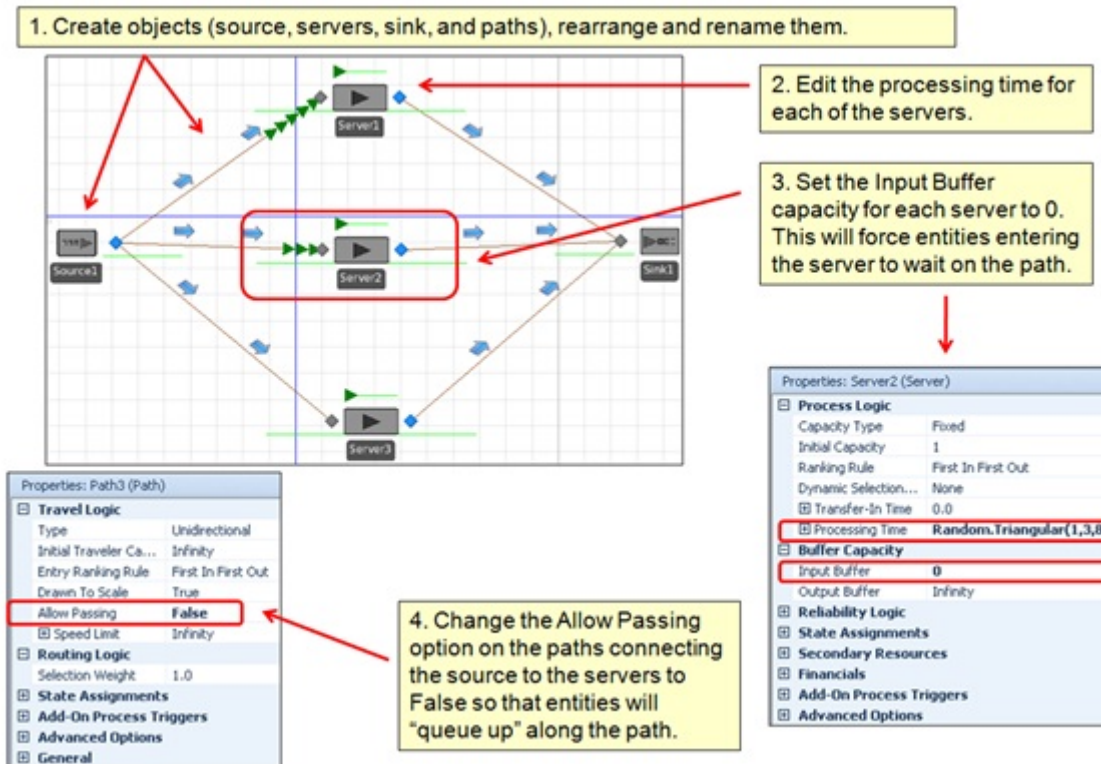
### 2.2. Creación del modelo – Objetos básicos

Los pasos para crear los objetos básicos del modelo son los siguientes (figura 6):

- 1) Crea, ordena, y renombra los objetos fuente, servidores, sumidero, y caminos.
- 2) Edita el tiempo de procesado de cada servidor según la descripción del sistema.
- 3) Especifica como 0 la capacidad del buffer de entrada de cada servidor. Esto hará que las entidades que intentan entrar en un servidor hayan de esperar en el camino hasta que el servidor esté libre.

4) En los caminos que conectan la fuente con cada servidor, asigna el valor de falso a la opción de permitir adelantamientos. De esta forma, las entidades tendrán que respetar la cola que se forme a lo largo del camino.

Figura 6. Creación del modelo – Objetos básicos



### 2.3. Creación del modelo – Lista de destinos

A fin de establecer una lista de nodos destino para las entidades que son generadas por la fuente, y lograr así que dichas entidades se dirijan al nodo destino con menor lista de espera, los pasos a dar son los siguientes (figura 7):

1) Selecciona la ventana de definiciones y, posteriormente, el icono lista. Selecciona ahora el icono nodo para crear una lista de nodos servidores hacia los que las entidades podrán encaminarse.

2) Selecciona el nodo de salida de la fuente. En la propiedad que establece la lógica de enrutamiento, cambia el tipo de entidad de destino de manera que esta pueda ser seleccionada desde una lista, especificando también el nombre de la lista creada en el punto anterior. Establece el criterio de selección como el de menor valor, y mantén la expresión de selección por defecto, *Candidate.Node.AssociatedStationOverload*.

Figura 7. Creación del modelo – Lista de destinos

1. To define a list of target nodes, click on the Definitions window, and then click on the List icon. Afterwards, click on the Node icon to create a node list of Server input nodes to select from routing entities.

2. Click on the output node for the Source. In the Routing Logic section change the Entity Destination Type to Select From List, and specify the Node List Name as the newly created node list. Next set the Selection Goal to Smallest Value and keep the default Selection Expression as: Candidate.Node.AssociatedStationOverload.

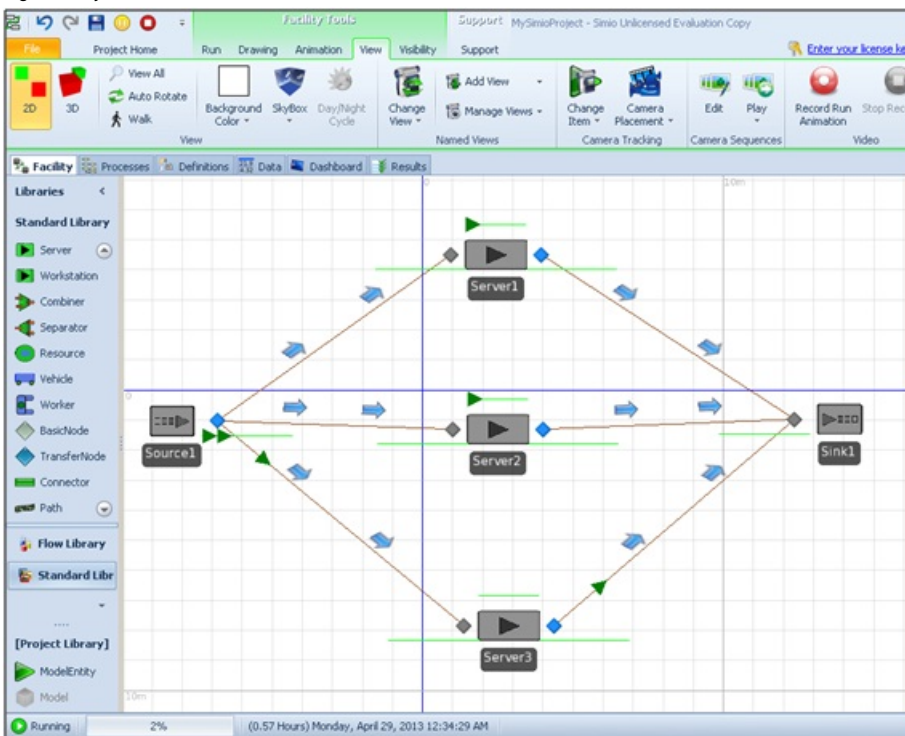
**Caution!** Although this generally balance the load between the three servers, we can further improve the balance by specifying the Blocked Routing Rule as Prefer Available.

Cabe notar que, si bien el procedimiento anterior permitirá balancear la carga entre los servidores de la lista, es posible mejorar dicho balanceo especificando la opción de preferir el nodo disponible en la propiedad *BlockedRoutingRule*.

### 2.4. Ejecución del modelo

Una vez creado el modelo, se procede a iniciar la simulación. El resultado debería ser similar al que se muestra en la figura 8. Observa que: (a) la velocidad con la que son atendidas las entidades en cada servidor se corresponde, aproximadamente, con los tiempos de procesamiento especificados en la descripción del sistema; y (b) durante la ejecución, las colas se mantienen balanceadas.

Figura 8. Ejecución del modelo

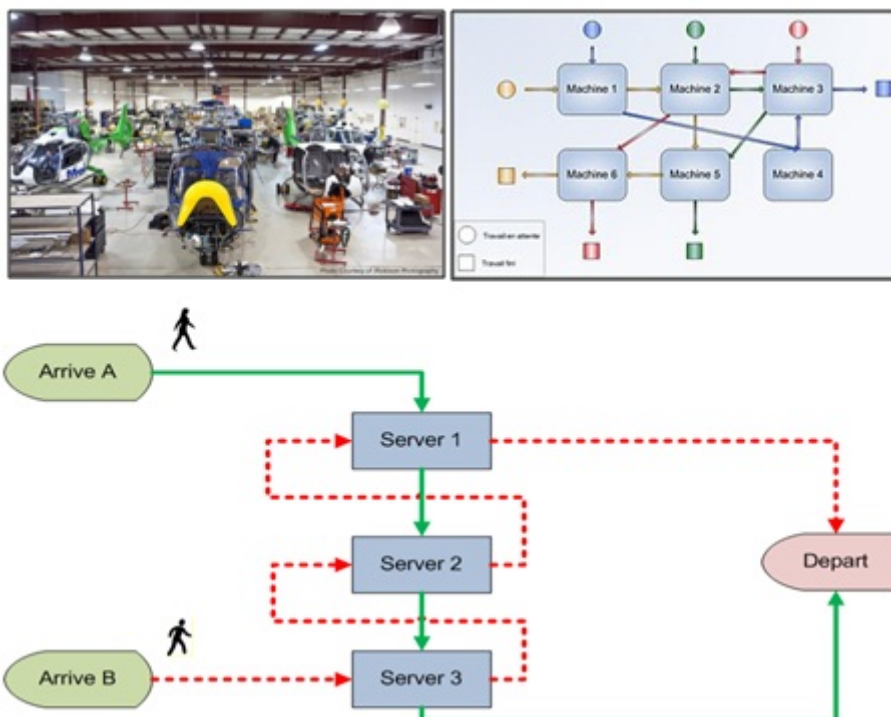


### 3. Modelado de tareas secuenciales (*Job-Shops*)

#### 3.1. Descripción del sistema

En determinados procesos productivos, es habitual que una entidad pase por diferentes etapas o máquinas (servidores) siguiendo una secuencia específica. Por ejemplo, en una fábrica cada tipo de pieza puede tener su propia secuencia de producción a lo largo de la cadena. Consideraremos dos tipos de entidades diferentes que se procesan a través de tres servidores distintos (figura 9). Las entidades de tipo A se procesan en la dirección *Servidor1* → *Servidor2* → *Servidor3*. Por contra, las de tipo B se procesan en la dirección inversa.

Figura 9. Representación gráfica del sistema



#### 3.2. Creación del modelo – Objetos básicos

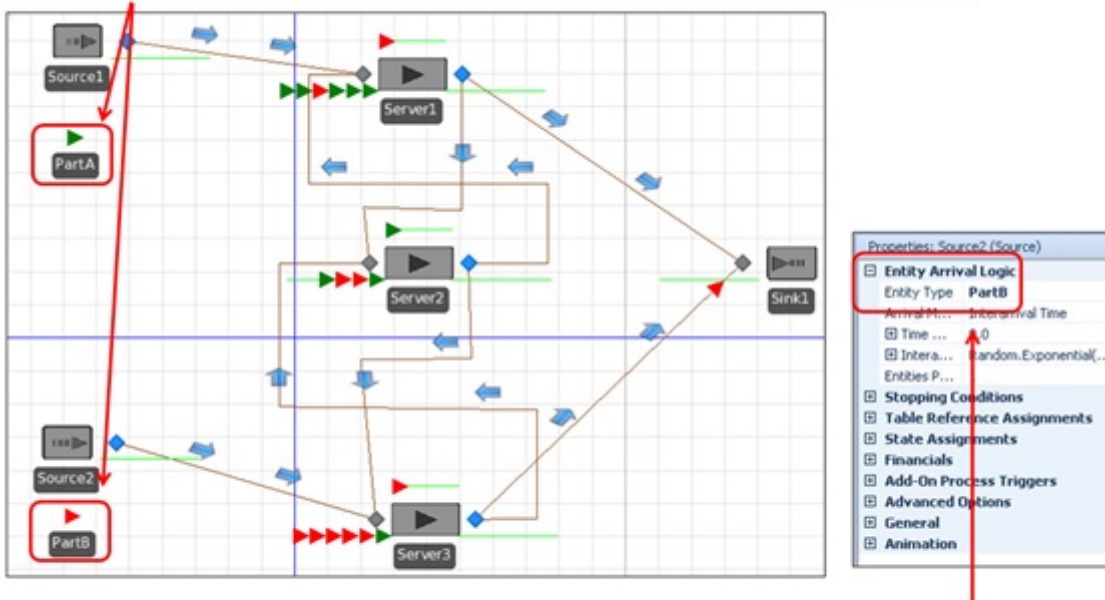
Los pasos para crear los objetos básicos del modelo son los siguientes (figura 10):

- 1) Crea, ordena, y renombra los objetos entidades (A y B), fuentes (1 y 2), servidores, sumidero, y caminos.

2) A fin de distinguir mejor los dos tipos de entidades, cambia el color de la de tipo B a rojo. Cambia el tipo de entidad que genera la fuente 2, de modo que esta fuente genere las de tipo B (por defecto, ambas fuentes generan instancias de tipo A).

Figura 10. Creación del modelo – Objetos básicos

1. Create objects (instances, source, servers, sink, and paths), rearrange and rename them.



2. Color Part B red and change the Entity Type property for the second Source to PartB.

### 3.3. Creación del modelo – Tablas de secuencia

El siguiente paso será definir la secuencia de actividades que ha de seguir cada entidad. Esto se hará mediante la creación de una tabla de secuencia por entidad (figura 11):

- 1) Selecciona la ventana de datos y, a continuación, la opción de añadir una tabla de secuencia. Cambia el nombre de la tabla para que haga referencia a la entidad A. Crea de forma análoga otra tabla de secuencia para la entidad B.
- 2) En la tabla de cada entidad, introduce el orden en que dicha entidad visitará cada uno de los servidores hasta llegar al sumidero (por ejemplo, en el caso de una instancia de tipo A sería: *Input@Server1, Input@Server2, Input@Server3, input@Sink1*).
- 3) Selecciona la entidad de tipo A y asígnale la tabla de secuencia correspondiente. Usa para ello la propiedad que define la secuencia inicial a seguir. Haz lo propio con la entidad de tipo B.
- 4) Para cada uno de los nodos de salida de las dos fuentes y los tres servidores, configura el tipo de destino para la entidad que lo atraviesa, de forma que este use las tablas de secuencia.

Figura 11. Creación del modelo – Tablas de secuencia

1. Click on the Data window and on Add Sequence Table. Rename the table PartASeq. Do the same with the second part and rename this second table PartBSeq.

2. Enter the sequence Input@Server1, Input@Server2, Input@Server3, Input@Sink for PartASeq. Enter the inverse sequence for PartBSeq.

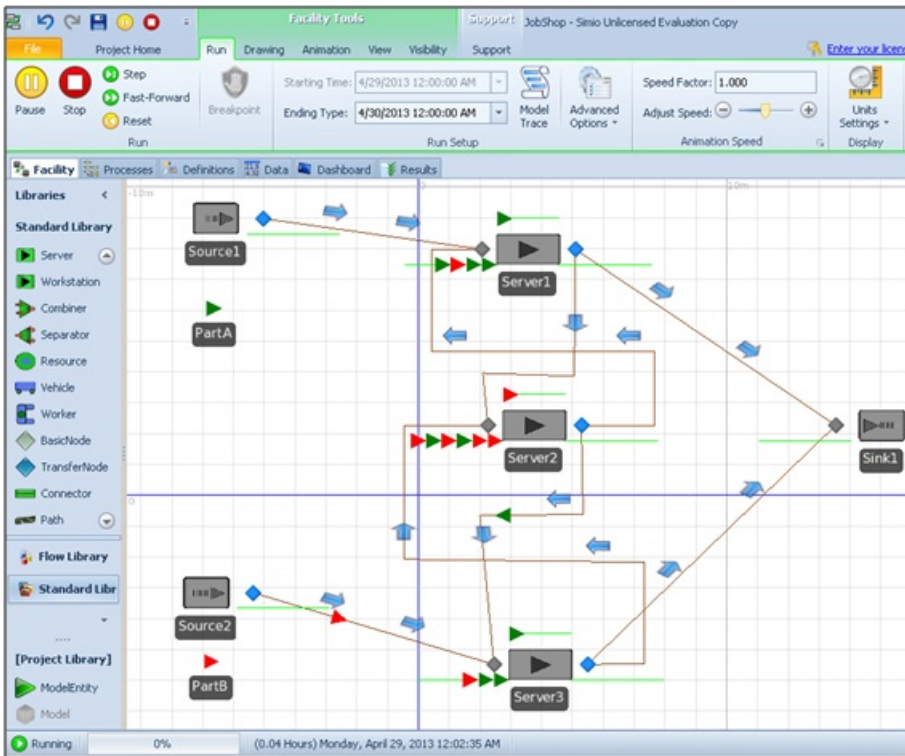
3. Click on the Part A entity and set its Initial Sequence to PartASeq. Do the same with Part B and PartBSeq.

4. Click on the output nodes for the two sources and three servers and specify Entity Destination Type as By Sequence.

### 3.4. Ejecución del modelo

Al ejecutar el modelo se debería observar cómo las entidades de tipo A salen de la fuente 1, mientras que las entidades de tipo B salen de la fuente 2 (figura 12). Ambos tipos de entidades siguen secuencias opuestas, cruzándose a lo largo de las mismas.

Figura 12. Ejecución del modelo







## Actividades

A continuación se incluyen una serie de actividades que permitirán afianzar los conocimientos adquiridos en este módulo. Todas las actividades son optativas excepto las que indique el/la profesor/a de la asignatura.

1. Desarrolla tus propios modelos Simio similares a los de este módulo.
2. Mejora la animación de los modelos usando tanto la biblioteca estándar como el repositorio 3D de Google.
3. Aplica alguno de los modelos a tu campo de interés.
4. Escribe un breve informe resumiendo las actividades anteriores. El informe debe contener: (a) una introducción al sistema (incluyendo un diagrama de flujo), (b) una descripción de cómo se ha desarrollado el modelo (incluyendo imágenes), (c) una sección experimental, (d) una discusión de los resultados, y (e) una conclusión.

## Bibliografía

**Joines, J. A.; Roberts, S. D.** (2012). *Simulation Modeling with SIMIO: A Workbook*. SIMIO LLC.

**Kelton, D.; Sadowski, R.; Swets, N.** (2009). *Simulation with Arena*. McGraw-Hill Science.

**Kelton, W.; Smith, J.; Sturrock, D.; Verbraeck, A.** (2010). *Simio and Simulation: Modeling, Analysis, Applications*. Learning Solutions.

**Simio** (2010). *Introduction to Simio*. <http://www.simio.com/about-simio/introduction-to-simio.htm>.

**Winter Simulation Conference Archive**. <http://informs-sim.org>.

