

Estrategias de control de líneas de espera

Estrategias de control de líneas de espera

Introducción	3
Estrategias de control de colas	3

Introducción

En este capítulo se muestra cómo modelar sistemas básicos de colas en una simulación de eventos discretos utilizando los bloques Cola de Entidades y Servidor de Entidades.

Como dijimos en capítulos anteriores, el bloque Cola de Entidades (línea de espera, LdE) almacena las entidades durante un tiempo que no se puede determinar de antemano. Un ejemplo cotidiano de cola es el de las personas que esperan en la cola de una caja registradora de una tienda. Un comprador no puede determinar de antemano cuánto tiempo debe esperar para completar su compra. Se puede utilizar la Cola de Entidades en diferentes aplicaciones, como los aviones que esperan para acceder a una pista de aterrizaje o los mensajes que esperan ser transmitidos. El bloque de Cola de Entidades tiene capacidad de almacenamiento, política de ordenación de entidades y política de sobrescritura de entidades. En función de estos parámetros, el bloque intenta dar salida a las entidades dependiendo de si el bloque posterior acepta nuevas entidades.

El bloque del Servidor de Entidades almacena las entidades durante un periodo de tiempo, llamado «tiempo de servicio», y luego intenta enviar la entidad dependiendo de si el bloque siguiente acepta nuevas entidades. Durante el periodo de servicio, el bloque está sirviendo la entidad que almacena. Un ejemplo cotidiano de servidor es una persona (como el cajero de un banco o de una tienda) con la que se realiza una transacción con una duración prevista.

Estrategias de control de colas

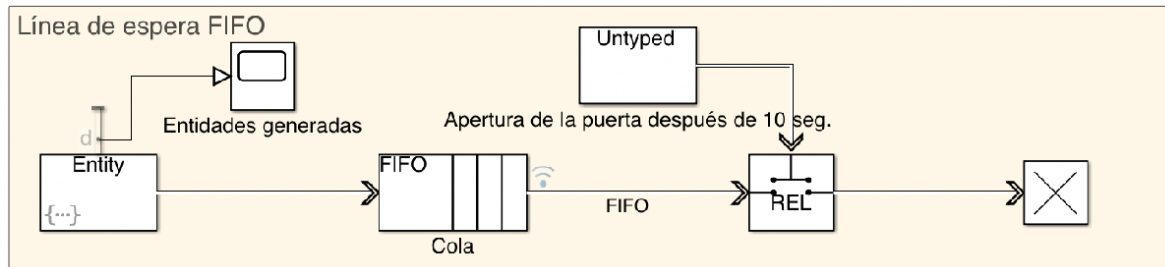
Se presentarán aquí diversos modelos que muestran cómo:

- Modelar una cola FIFO, una cola LIFO y una cola prioritaria.
- Especificar políticas de sobrescritura de entidades cuando la cola alcanza su capacidad.

- Personalizar y variar los tiempos de servicio de las entidades.
- Asignar y cambiar los atributos de las entidades en función de los eventos.
- Explorar la longitud de una cola

FIFO, LIFO y prioridad

El modelo para una LdE LIFO es el siguiente:



En el bloque generador de entidades se introduce el siguiente código, que genera 10 entidades con 1 segundo de intervalo entre ellas (**Time source - MATLAB action**):

```
% Patrón de repetición de secuencia
% Se inicializan las variables
persistent SEC;
persistent idx;
if isempty(SEC)
    % Se generan 10 entidades con 1 segundo de intervalo
    SEC = [1 1 1 1 1 1 1 1 1 1];
    idx = 1;
end
if idx > numel(SEC)
    % Se detiene la generación de entidades después de 10
    dt = inf;
else
    dt = SEC(idx);
end
idx = idx + 1;
```

El bloque genera una entidad y especifica el atributo `Atributo1` en cada entidad. Se pueden utilizar atributos para representar características o propiedades de las entidades. En este ejemplo, la primera entidad lleva un valor de 1, y el valor del atributo de cada entidad generada se incrementa en 1. Para lograr este comportamiento, en el bloque **Generador de Entidades**, en la pestaña **Acciones de Eventos**, en el campo **Generar acción**, se utiliza este código:

```
% Se modifica el valor del atributo de la entidad en cada paso de reloj.
persistent SEC1;
persistent idx1;
```

```

if isempty(SEC1)
    SEC1 = [1 2 3 4 5 6 7 8 9 10];
    idx1 = 1;
end

if idx1 <= numel(SEC1)
    entity.Atributo1 = SEC1(idx1);
end
idx1 = idx1 + 1;

```

Las entidades generadas son enviadas al bloque de Cola de Entidades. Para ver en qué orden salen los elementos de la cola, es decir, cómo se comporta la clasificación, el bloque de Cola de Entidades está conectado a una Puerta de Entidades (**Entity Gate**) configurada como puerta de liberación (**Operating mode - Release Gate**). Una puerta de liberación permite el paso de una entidad cuando recibe una entidad con un valor positivo (mayor que 0) desde su puerto de control. Las puertas bloquean las entidades durante los primeros 10 segundos y las almacenan en la cola. Después de los primeros 10 segundos, las puertas permiten el paso de una entidad por segundo en función de la política de clasificación. El código para el generador de entidades que gobiernan el puerto de control de la Puerta de entidades es el siguiente:

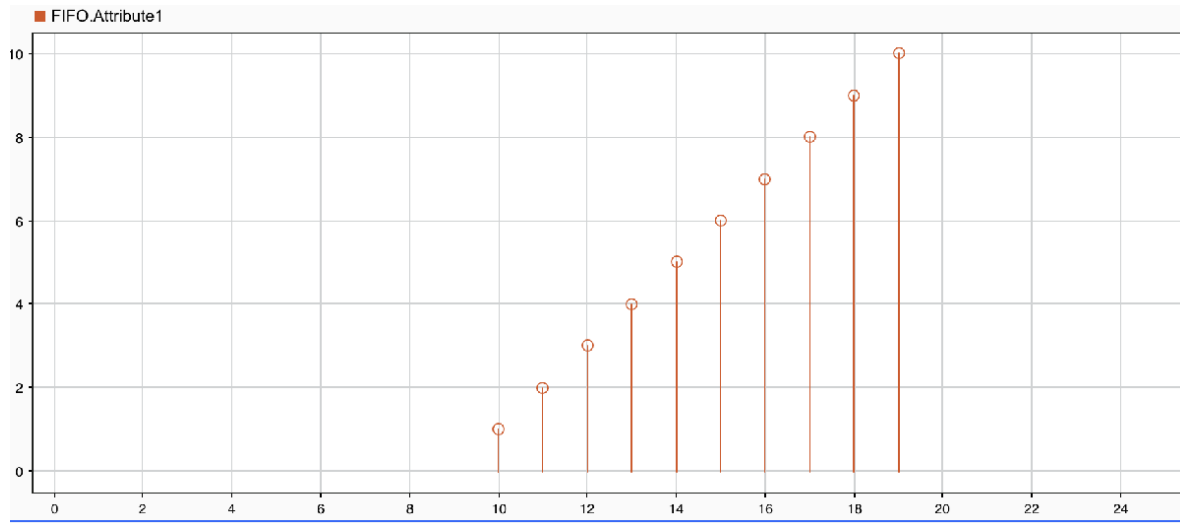
```

% Cada valor de la secuencia representa el retardo en liberar las entidades
% de la cola
persistent SEC;
persistent idx;
if isempty(SEC)
    SEC = [10 1 1 1 1 1 1 1 1 1];
    idx = 1;
end
if idx > numel(SEC)
    dt= inf;
else
    dt = SEC(idx);
end

idx = idx + 1;

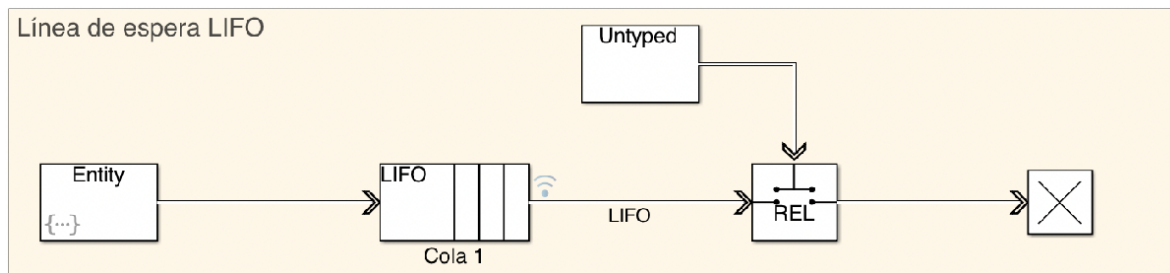
```

Si ejecutamos la simulación y monitorizamos la cola FIFO (usando **Data Inspector**), observamos la siguiente figura:

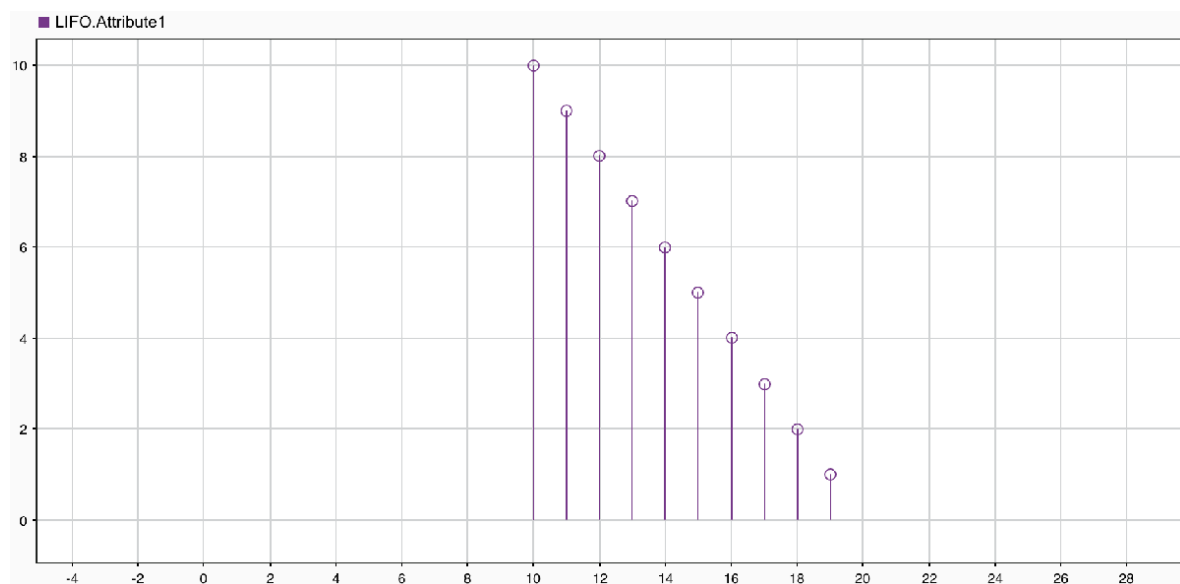


donde vemos que las entidades (cada una con su valor) no son liberadas hasta el instante 10 y conservan el orden de creación.

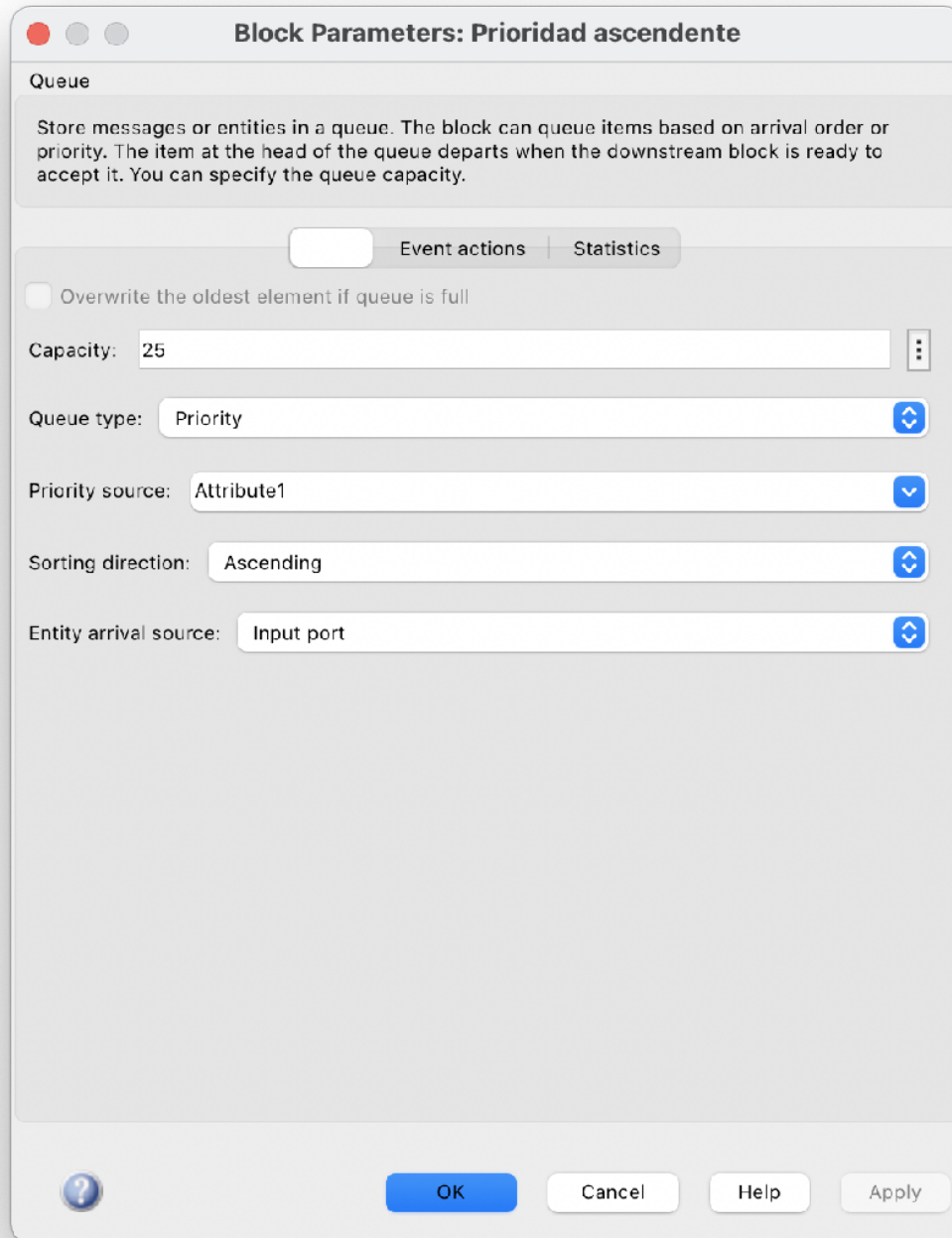
El esquema para la LdE LIFO es el mismo que para FIFO:



Todo el código es el mismo, la diferencia es que en el tipo de cola del bloque Cola 1 hemos elegido como tipo LIFO. Su gráfica de salida se muestra a continuación:

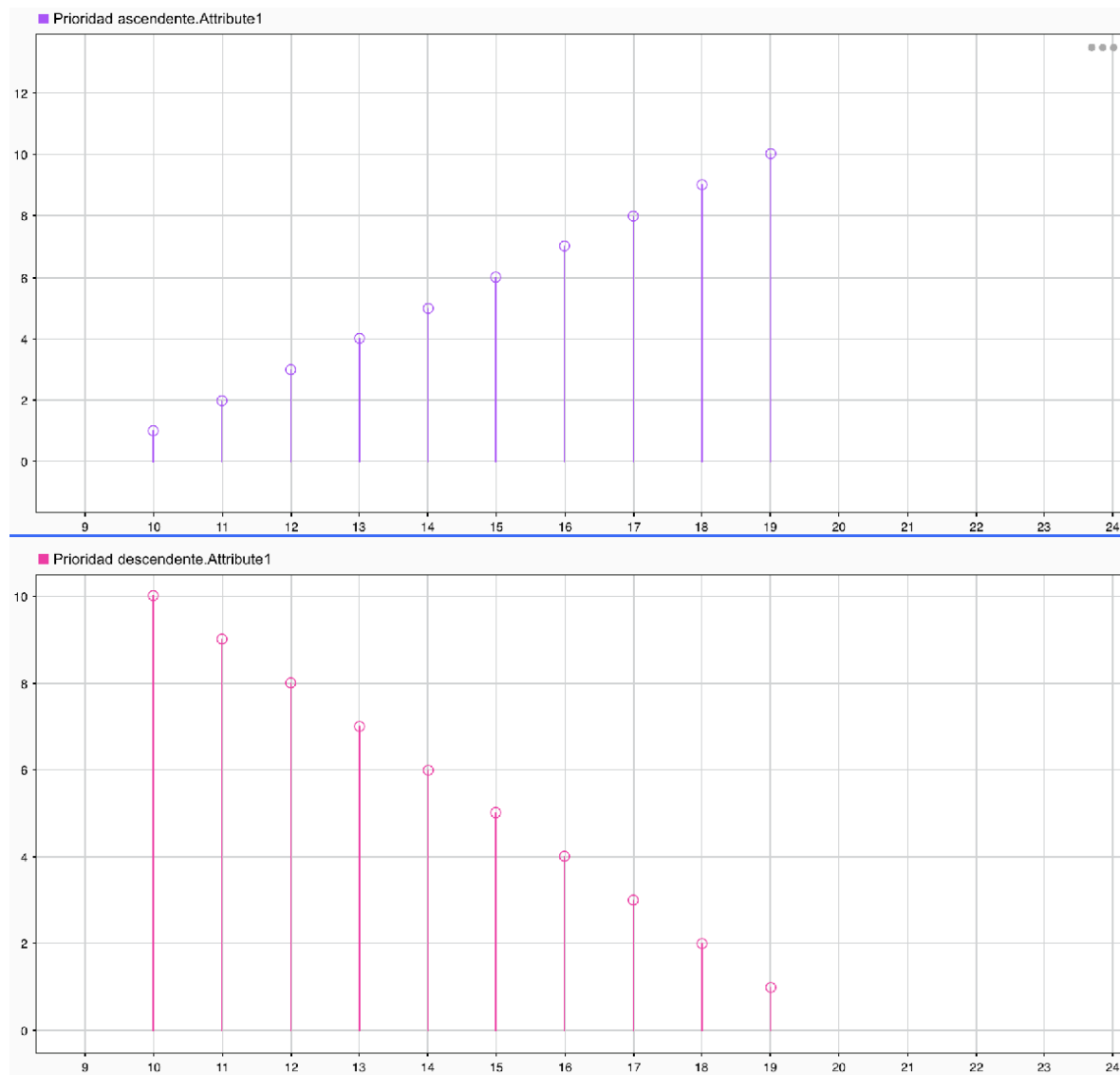


Para las líneas de espera con prioridad sólo es necesario operar sobre el bloque de cola. Para ello, usamos su ventana de configuración:



En ella, establecemos el campo **Queue type** como **Priority** y **Sorting direction** como ascendente o descendente. Los resultados pueden observarse en las siguientes gráficas

obtenidas con **Data Inspector**:



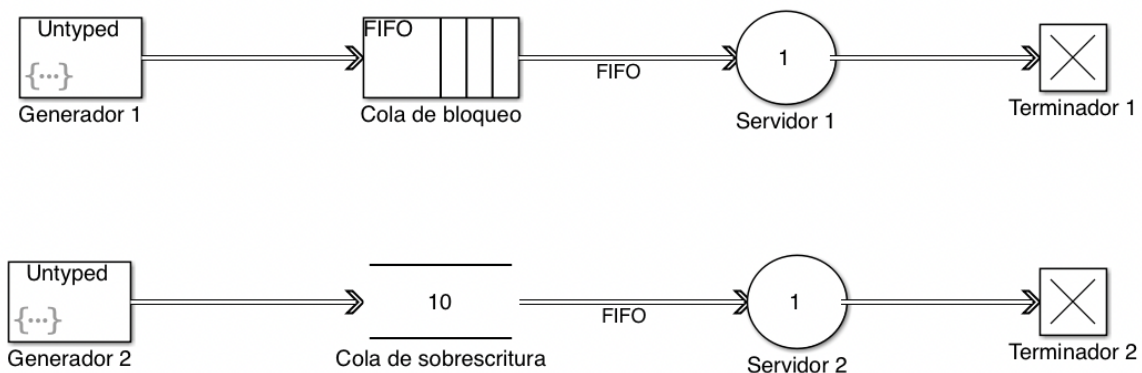
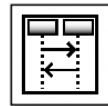
Bloqueo y sobrescritura de colas

Una de las estrategias de control de colas se puede establecer mediante el bloqueo de la misma o la sobrescritura de los elementos de la cola por aquellos que se incorporan a la misma. Ambas estrategias se pueden especificar cuando el bloque llega a su capacidad máxima especificada. Esta política se especifica mediante la casilla **Overwrite the oldest element if queue is full** (Sobrescribir el elemento más antiguo si la cola está llena). Entonces, si dicha casilla...

- ... está desactivada, la cola no admite nuevos elementos cuando se llega a su máxima capacidad y la cola se bloquea.
- ... está activada, la cola siempre acepta una entidad entrante sobrescribiendo la

entidad más antigua. El bloque sobrescribe la entidad más antigua, pero la entidad que sale del bloque está determinada por la política de ordenación de la cola.

Para ver cómo se comportan ambas estrategias, vamos a construir dos modelos idénticos con políticas de bloqueo diferentes en la cola: una permitirá sobrescribir entidades en cola, y otra tendrá un tamaño limitado y no permitirá la sobrescritura. En ambos modelos, el servidor está establecido con un tiempo de servicio fijo (25 segundos) y las colas podrán almacenar un máximo de 10 entidades, generándose 1 por segundo. Por último, añadiremos un bloque **Sequence Viewer** para analizar el comportamiento de ambos modelos:



Configuramos los bloques de la siguiente manera:

Generador 1 y Generador 2: en la pestaña **Entity type**, en el campo **Entity type** lo establecemos a **Anonymous**. En la pestaña **Event actions**, en el evento **Generate**, escribimos el siguiente código: `entity = rand();` con este código lo que generamos es un número aleatorio en cada segundo (establecido en la pestaña **Entity generation - Period**).

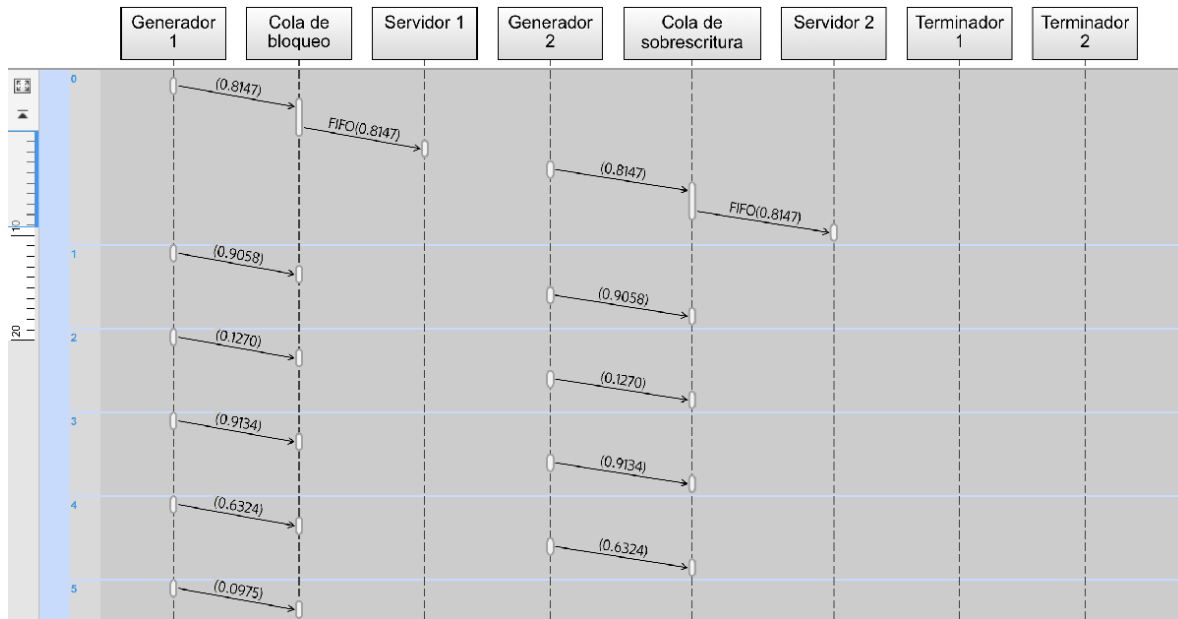
Cola de bloqueo: En la pestaña **Main** ponemos 10 como capacidad máxima de la cola (**Capacity**) y nos aseguramos que la casilla **Overwrite the oldest element if queue is full** está desactivada.

Cola de sobrescritura: En la pestaña **Main** ponemos 10 como capacidad máxima de la cola (**Capacity**) y nos aseguramos que la casilla **Overwrite the oldest element if queue is full** está activada.

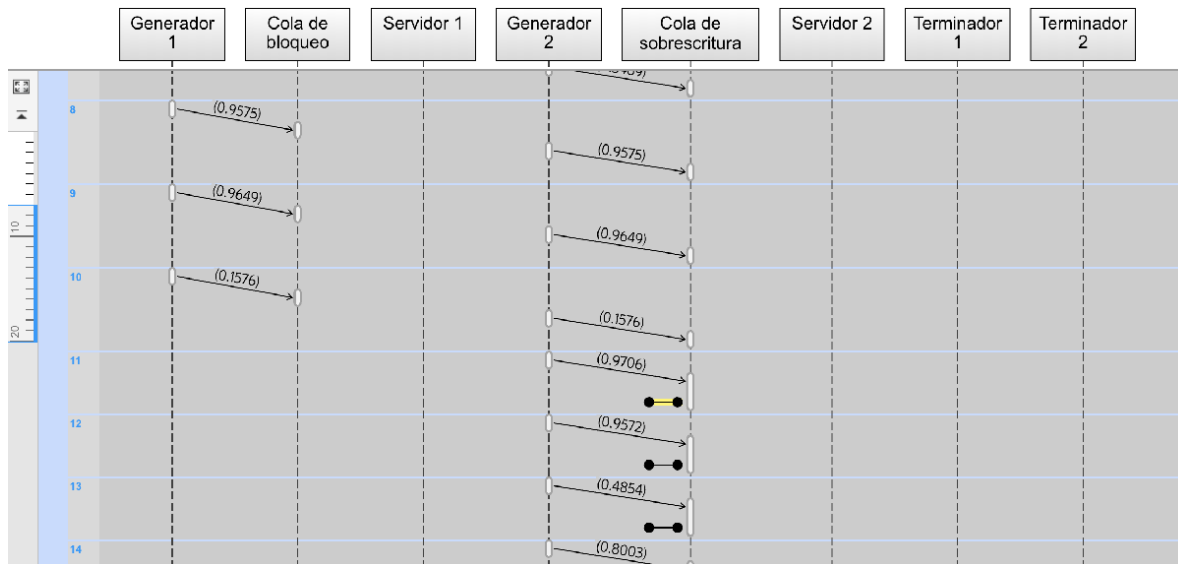
Servidor 1 y Servidor 2: En ambos establecemos un tiempo de servicio (**Service time value**) de 25 (segundos).

Si ejecutamos la simulación y hacemos doble click sobre el bloque **Sequence Viewer**,

obtendremos una figura como la siguiente:



En ella podemos observar que la secuencia de generación empieza en el instante 0 con la entidad 0.8147 que pasa a la *cola de bloqueo* y seguidamente al *servidor 1* donde permanecerá 25 segundos (el tiempo de servicio). El *generador 2* genera el mismo elemento y se procede de la misma manera. Durante los próximos 10 segundos se irán generando entidades, y en el instante 11 pasarán dos cosas: la primera, que la *cola de bloqueo* no admite más entidades para la producción de las mismas. Y la segunda, la *cola de sobrescritura* sigue admitiendo entidades sobrescribiendo las entidades que más tiempo llevan en la cola. Lo vemos en la siguiente figura:



En esta figura, vemos que se genera el elemento 0.9706 y sobrescribe al elemento 0.9058

que fue el segundo en ser generado. Se puede pulsar sobre la marca de sobrescritura (dos punto unidos por una barra) para ver qué elemento de la cola se ha sobrescrito.

Establecer los tiempos de servicio

Existen varias estrategias para establecer tiempo de servicio en los servidores. Todas ellas se gobiernan desde el campo **Service time source** de la pestaña **Main** del bloque servidor. Dichas estrategias son las siguientes:

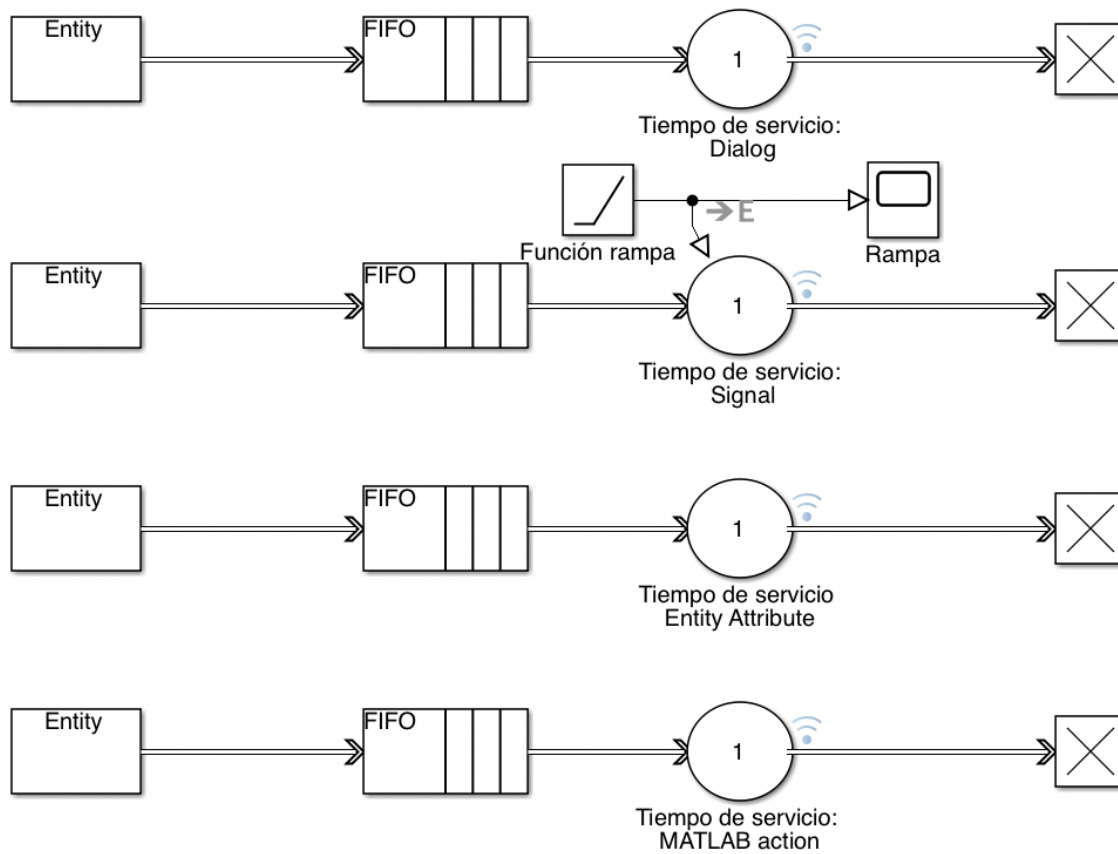
Dialog - Establece un tiempo fijo de servicio que se establece en el campo **Service time value**.

Signal port — Se establece una señal de entrada desde un bloque de Simulink. Puede ser cualquiera señal discreta.

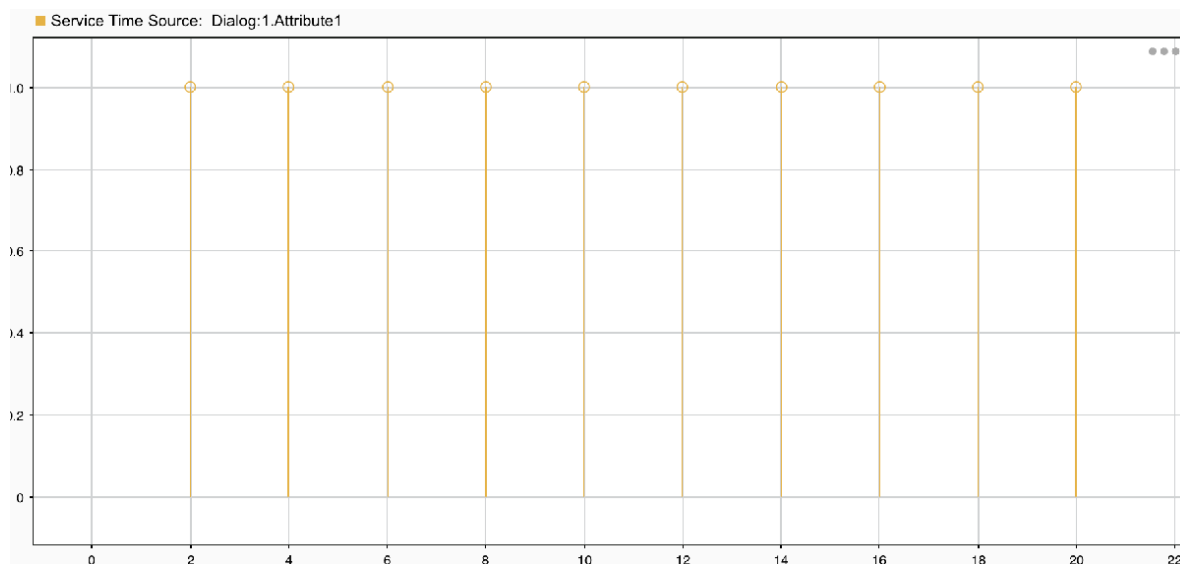
Attribute — El tiempo de servicio se establece en un atributo en el bloque generador. Esto permite que cada una de las entidades que se «mueven» en el sistema tengan su tiempo asociado.

MATLAB action — Permite introducir un código que establezca el tiempo de servicio en el campo **Service time action** y lo asigna a la variable establecida **dt** que es el parámetro que el modelo utiliza como tiempo de servicio. Por ejemplo, **rand(1, 1)** genera un valor de servicio entre 0 y 1 uniformemente distribuido.

El modelo que establece un ejemplo de cada uno de las cuatro configuraciones anteriores es el siguiente:



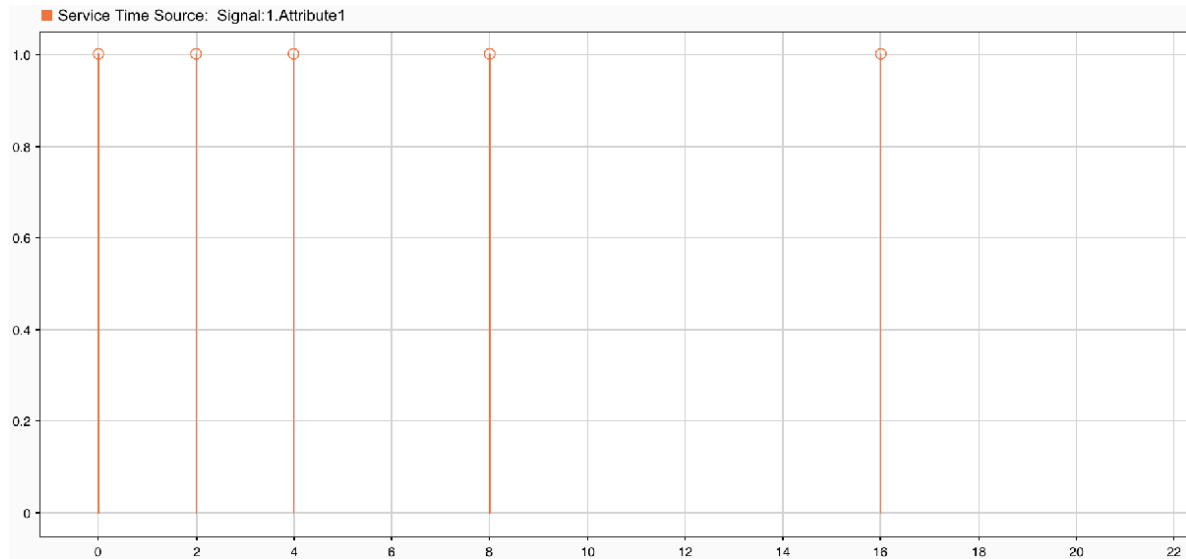
Si utilizamos Data inspector para ver cómo es la generación y servicio, observamos lo siguiente. Para el primer modelo:



Al estar establecido el tiempo de servicio en 2, la primera entidad llega en el tiempo 0 y permanece 2 segundos en servicio, siendo liberada en el tiempo 2. La segunda entidad

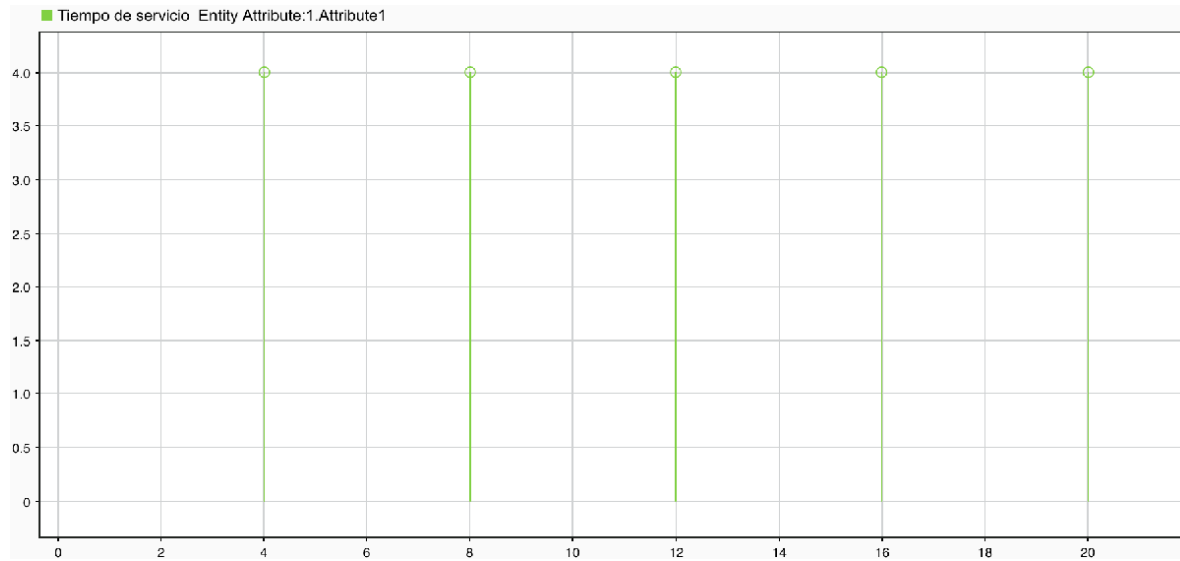
se genera en el tiempo 1 y permanece a la espera a que pueda ser atendida en el servidor que está ocupado con la primera entidad. En el tiempo 2 el servidor se libera y empieza a atender a la segunda entidad liberándola en el tiempo 4 tras 2 segundos de servicio y así sucesivamente.

Para el segundo modelo:



La entidad que se genera en el tiempo 0 es inmediatamente atendida pues la función rampa establece un tiempo de 0 para ella. La entidad que se genera en el momento 1 pasa a ser atendida en el momento 1, y tiene una duración de servicio de 1 segundo, liberándose en el tiempo 2. La entidad que se genera en el momento 2 pasa a ser atendida, pues ya ha acabado la anterior y tiene un tiempo de servicio establecido por la función rampa de 2, por tanto, es liberada en el momento 4. La entidad que se genera en el tiempo 3 no puede ser atendida en dicho momento pues el servidor está ocupado en el proceso de la entidad anterior. Cuando puede empezar a ser procesada, instante 4, su duración de servicio será de 4 según establece la función rampa, siendo liberada en el instante 8, y así sucesivamente. (Nota: observará que hay una pequeña «E» en la señal que sale del bloque rampa al servidor. Esta «E» se pone automáticamente al detectar que la señal debe ser conducida como un evento y no como una señal guiada por el tiempo. Se actualizará automáticamente cuando se ejecute la simulación por primera vez).

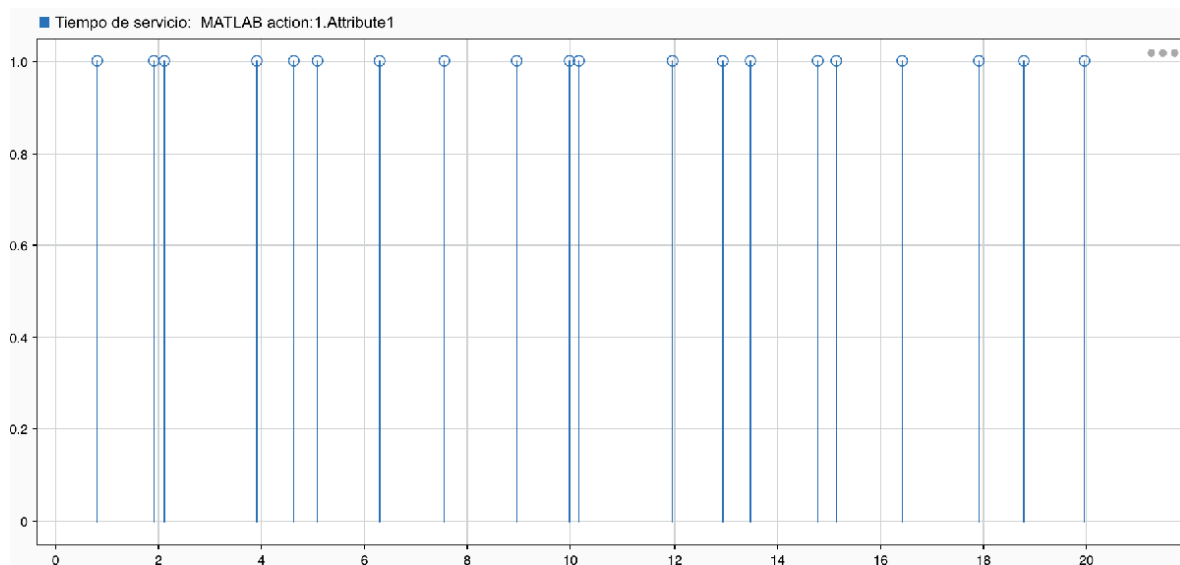
Para el tercer modelo:



En el bloque generador se establece (en **Entity type**) un atributo con valor de 4. En el bloque servidor se indica que el tiempo de servicio sea establecido por el valor de dicho atributo. Por tanto, como se ve en la gráfica, cada entidad tiene un valor de 4 y un tiempo de servicio de 4.

Y para el cuarto modelo:

El tiempo de servicio en este último modelo se establece de forma aleatoria (entre 0 y 1). Su gráfica de servicio es:

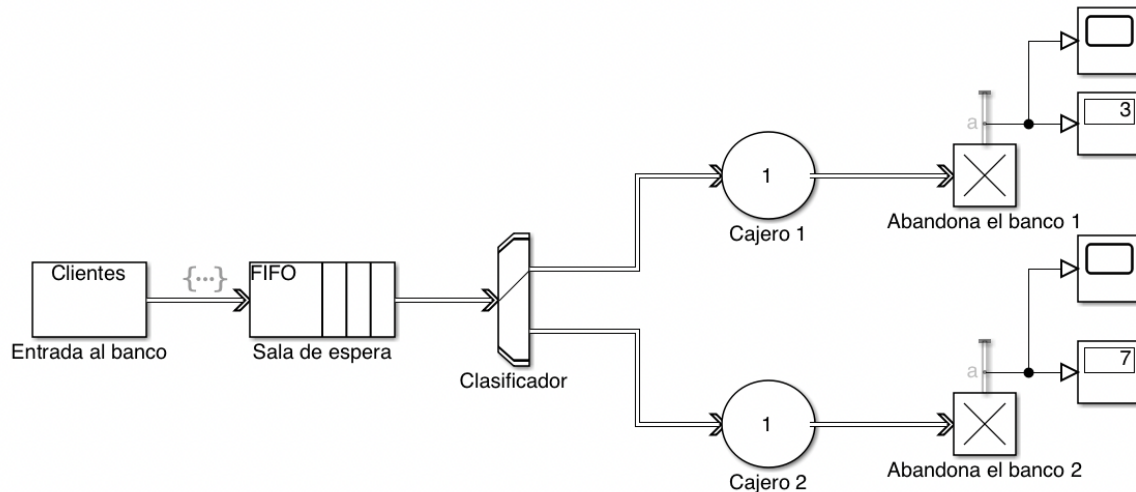


Selección de servicio según el valor de un atributo

Se pueden establecer atributos a las entidades para representar sus características o su futuro comportamiento. En un sistema de LdE, se pueden utilizar acciones como respuestas a eventos y cambiar los atributos de las entidades. Por ejemplo, puede cambiar

el valor de un atributo de entidad cuando la entidad entra y sale del bloque de Cola de Entidades. En el bloque de Cola de Entidades, en la pestaña de **Event actions**, se puede ver el conjunto de eventos para los que puede crear acciones.

Supongamos que queremos modelar un sistema de atención al cliente en una sucursal bancaria. La sucursal tiene dos cajeros, cada uno asignado a un tipo de transacción particular. Los clientes llegan a la sucursal. Eligen un número para su transacción y son dirigidos al cajero correcto. Los clientes abandonan la sucursal una vez finalizada la transacción. El modelo es el siguiente:



Para configurar los diferentes bloques, haremos lo siguiente:

- En el **Clientes**, haremos doble-click e iremos a la pestaña **Entity type** y definiremos un nuevo atributo al que llamaremos **TransaccionTipo** con un valor inicial de 0. Todos los demás valores los dejamos por defecto. Este atributo del cliente es el que utilizaremos para determinar qué tipo de cliente es.
- En el bloque de LdE **Sala de espera**, iremos a la pestaña **Event actions** e introduciremos en el evento **Entry** el siguiente código : `entity.TransaccionTipo = randi([1 2]);` en el que asignamos a la entrada en la cola un valor entero aleatorio entre 1 y 2 que nos ayudará a discriminar qué tipo de cliente es para dirigirle a un cajero o a otro. Todos los demás parámetros se dejan por defecto.
- En el bloque **Clasificador (Entity output switch)**, en el campo **Switching criterion** elegiremos **from attribute** y seleccionamos en **Switch attribute name** el atributo **TransaccionTipo**. Los demás parámetros los dejamos por defecto.

Ejecute el modelo y observe los resultados.

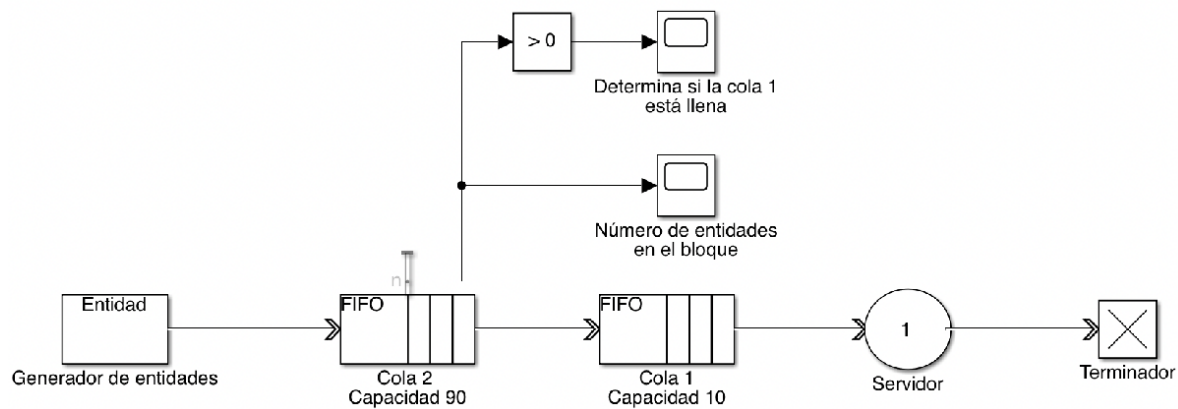
Partición de una cola para explorar su longitud

Supongamos que queremos determinar qué proporción de tiempo la longitud de la cola excede de 10 para una cola con una capacidad de 100. Esto se puede investigar

utilizando un par de colas conectadas en serie. Las colas tienen longitudes de 90 y 10. Juntas representan una cola con una capacidad de 100.

La partición de la cola original en dos colas más pequeñas nos permite recopilar estadísticas relacionadas con una de las colas más pequeñas. Por ejemplo, puede ver la estadística de la longitud de la cola para el bloque de la cola de entidades con una capacidad de 90. Si hay entidades acumuladas en la cola con capacidad de 90, significa que la cola con capacidad de 10 está llena. Por lo tanto, determinar la proporción de veces que la cola con capacidad de 100 tiene un mínimo de 10 entidades equivale a determinar la proporción de veces que la longitud de cola de la cola con capacidad de 90 es mayor que 0.

Lo vemos con el modelo siguiente:



Para configurar los bloques, hacemos lo siguiente:

- En el bloque **Entidad**, seleccionaremos **MATLAB-action** en **Time source** e introduciremos el siguiente código:

```
persistent rngInit;
if isempty(rngInit)
    seed = 123456;
    rng(seed);
    rngInit = true;
end
```

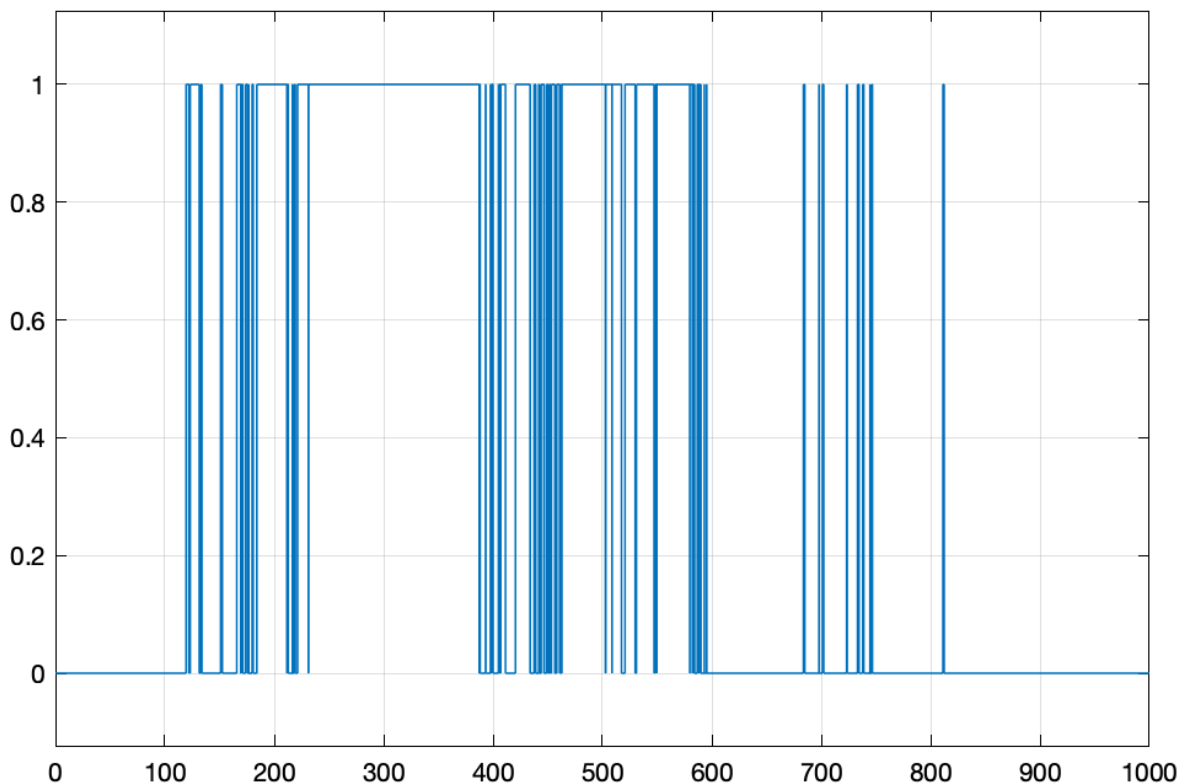
```
% Distribución uniforme entre
% m: Mínimo, M: Máximo
m = 0; M = 5;
dt = m + (M - m) * rand;
```

en este código, inicializamos una secuencia de números aleatorios con una determinada semilla para que en las repeticiones tengamos siempre la misma secuencia y luego la secuencia configura una serie entre 0 y 5, que es el intervalo de tiempo en el que se generan entidades seleccionado de forma aleatoria. Todos los demás parámetros se

dejan por defecto.

- El bloque **Cola 2** se establece con una capacidad de 90 y en la pestaña **Statistics** seleccionamos **Number of entities in block**
- El bloque **Cola 1** se establece con una capacidad de 10. Todos los demás datos se dejan por defecto.
- En el bloque **Servidor**, se procede de la misma manera que con el bloque **Entidad** y se utiliza el mismo código, de manera que el tiempo de servicio sea igual al tiempo de generación y no se produzcan atascos por servicio.

En el visor de la Cola 2 vemos lo siguiente:



Aquí se perciben los instantes en los que la **Cola 1** ha sido desbordada. En el visor de entidades en el bloque vemos:

