

# Simulación de Sistemas. Ejercicio 2

Antonio Jesús Heredia Castillo

---

## 1. Modelo diseñado

Para este ejercicio, me he basado en el sistema creado en practicas de colas en un servidor (colmmk). La diferencia de ese ejemplo y el que estamos usando en este ejercicio es uno básico. Y es que en aquel ejemplo solo pasaba por un servidor. En este ademas de que pasa por dos servidores, primero por A y luego por B, cada uno tiene distintos tipos de servicio. Como se puede pensar de forma intuitiva, en este ejercicio vamos a necesitar un suceso mas, al que se llega después de pasar por el segundo servidor.

## 2. Grafo de sucesos

Los suceso que tendrá mi sistema son:

1. **suceso\_llegada:** Es por el primer suceso que pasa cuando llega al servidor. Y se esta esperando a ser atendido por el servidor A.
2. **suceso\_fin\_a:** Cuando termina en el servidor A y empieza a esperar (o no) a ser servido por B.
3. **suceso\_fin\_b:** Termina su paso por el servidor B.
4. **suceso\_finsimulacion:** Cuando termina la simulación.

El grafo de suceso quedara de la siguiente forma: Las constantes tendrán las siguientes

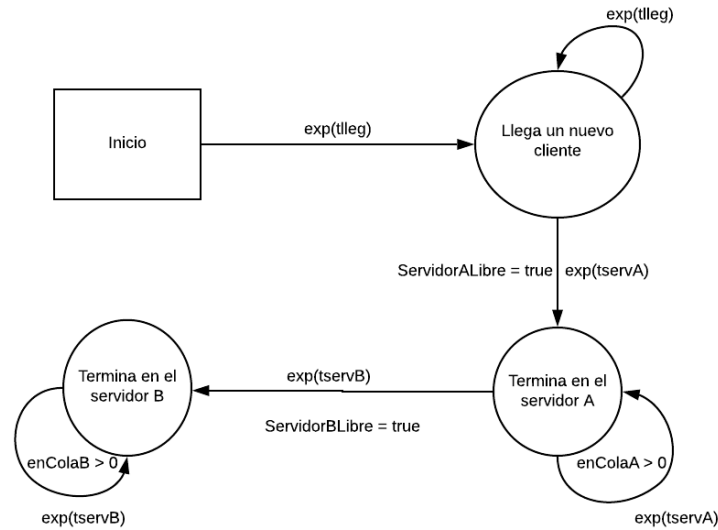


Figura 1: Grafo de sucesos

valores como se describen en el enunciado del ejercicio.

- $tlleg = 1,0$
- $tservA = 0,8$
- $tservB = 1,2$

### 3. Detalle de las rutinas y variables mas destacadas

Tenemos varias constantes para el uso de nuestro sistema:

- **suceso\_llegada**: definición del tipo de suceso de llegada.
- **suceso\_fin\_a**: definición del tipo de suceso de fin del uso del servidor A.
- **suceso\_fin\_b**: definición del tipo de suceso de fin del uso del servidor B.
- **suceso\_finsimulacion**: definición del tipo de suceso de finalización de la simulación.

En el programa usamos varias variables para poder llevar el sistema de simulación. A continuación detallo las mas importantes:

- **Actual** suceso actual en la simulación.
- **enColaServA** cantidad de clientes que hay en la cola del servidor A.
- **enColaServB** cantidad de clientes que hay en la cola del servidor B.
- **libreA** si el servidor A esta libre.

- **libreB** si el servidor B esta libre.
- **nClientAntent** el numero total de clientes atendidos.
- **tTotalEnSistema** el tiempo total que han pasado los clientes en el sistema.
- **lsuc** lista de sucesos.
- **tLlegadas** lista de tiempos de llegadas.

Por no añadir código en la memoria, pondré el nombre de las funciones mas importantes que uso en el programa y daré una pequeña explicación de cada una.

La función "void suceso()."es la que con un switch-case, decidirá, según el suceso Actual.

La función "void iniciar\_variables()."es donde iniciaremos todas las variables con los siguientes valores:

- **reloj** = 0.
- **tParada** = 0.
- **enColaServA** = 0.
- **enColaServB** = 0.
- **libreA** = true.
- **libreB** = true.
- **nClientAntent** = 0.
- **tTotalEnSistema** = 0.
- **lsuc.clear()**.
- **tLlegadas.clear()**.

La función "void suceso\_fin()"sera la encargada de finalizar la simulación. Se modifica la variable de finalización y ademas se calcula el tiempo medio de esa ejecución añadiendolo a la lista de tiempos medios.

La función "void suc\_fin\_b()"sera la función que se ejecute cuando el suceso finalice en el servidor B. Aquí añadimos uno al numero de clientes que han pasado por el sistema, sumamos el tiempo que el cliente ha pasado en el sistema. Ademas si queda algún cliente en la cola del servidor B, añadimos un nuevo suceso de salida del servidor B. Si no hay clientes en cola indicamos que el servidor esta libre.

La función "void suc\_fin\_a()"si hay clientes en la cola del servidor A, se genera un nuevo suceso para que pase el siguiente cliente, si no hay nadie se indica que el servidor esta libre. En caso de que el servidor B este libre se pasa el cliente al servidor B en caso de que no este libre se añade un nuevo cliente a la cola de ese servidor.

La función "void llegaServ()." es la que se ejecuta con el suceso de llegada de un nuevo cliente. Lo primero que se hace es añadir el tiempo de llegada de ese cliente al servidor para luego calcular el tiempo de estancia y se añade ese suceso. Si el servidor A ya está libre, se inserta el suceso de fin del servidor A que equivale a que el cliente ya ha entrado y terminado en ese servidor. Si no está libre se añade un nuevo cliente a la cola del servidor.

Además tenemos varias funciones para el uso simplificado del programa. Además también está la función de generador exponencial usado en la práctica 3 en el código `colmmark`.

La función que uso para mostrar el informe simplemente suma todos los tiempos medios y calcula la media de todas las simulaciones realizadas.

## 4. Experimentación

Lo primero que he realizado era ver que resultado nos daba la ejecución del sistema con las condiciones iniciales. Usando una sola simulación como podemos ver en la siguiente tabla, da resultados muy dispares.

Nº Ejecución	Tiempo medio de espera
1	53
2	43
3	29
4	62
5	73

Como suele pasar en este tipo de problemas realizando un número de simulaciones más elevado (en este caso 1000) suele dar unos valores menos dispares como podemos ver en la siguiente tabla.

Nº Ejecución	Tiempo medio de espera
1	46.32
2	46.23
3	45.544
4	45.961
5	46.597

Ahora realizaremos varias pruebas para ver cómo afecta la disminución del **t<sub>servB</sub>** hasta que baje de los 10 minutos el tiempo medio de espera.

t <sub>ServB</sub>	Tiempo medio de espera
1.2	46.244
1.1	31.773
1	18.545
0.9	10.139
0.85	8.289

Como con un valor de **0.9** aunque es por muy poco, todas las ejecuciones que he realizado están por encima de 10 minutos, para asegurar que el tiempo medio de espera sea inferior a 10 minutos podemos poner un **t<sub>Serv</sub>=0.85**.