

# Informe: Proyecto Simulación I

Integrantes:

Davier Sánchez Bello C-412

Maykol Luis Martínez Rodríguez C-412

February 28, 2024

# 1 Introducción

En este proyecto simularemos un sistema de eventos discreto, más específicamente un sistema donde un grupo de servidores, colocados en serie, atienden a clientes que tienen la necesidad de pasar por cada uno de los servidores en el orden en que aparecen. La llegada de los clientes así como el tiempo de atención en cada uno de los servidores están dados por variables aleatorias de diferente distribución, especificadas al comienzo de la Simulación.

## 1.1 Objetivos y metas

Queremos diseñar una estructura que permita simular el sistema para cualquier combinación de distribuciones. De esta manera, aquellos que intenten implementar sistemas similares en la vida real podrán tomar decisiones de una manera informada y estudiar cuál es la mejor manera de colocar los servidores para minimizar, ya sea el tiempo de espera en colas de los clientes, la cantidad de colas que estos hacen, o la longitud en promedio o máxima de las colas, o el tiempo de inactividad de los servidores. Es nuestra meta, lograr proveer una simulación que se adapte bien para casos de la vida real con características similares a nuestra simulación de servidores en serie.

## 1.2 Descripción del sistema

A continuación la descripción del sistema tal y como aparece en la orden:

Los clientes llegan a un sistema que tiene  $n$  servidores, y las llegadas distribuye  $M$ . Cada cliente que llega debe ser atendido primero por el servidor 1 y, al completar el servicio en el servidor 1, el cliente pasa al servidor 2. Cuando un cliente llega, entra en servicio con el servidor 1 si ese servidor está libre, o se une a la cola del servidor 1 en caso contrario. De manera similar, cuando el cliente completa el servicio en el servidor 1, entra en servicio con el servidor 2 si ese servidor está libre, o se une a su cola y así sucesivamente. Después de ser atendido en el servidor  $n$ , el cliente abandona el sistema.

## 1.3 Variables de Interés

Estamos interesados en el tiempo de espera promedio por cliente, los tiempos máximo y promedio de inactividad de los servidores, la mayor longitud

alcanzada por la cola de cualquier servidor, así como la longitud promedio en la cola de cada servidor.

## 2 Como correrlo

Para usar una interfaz visual un pelín más atractiva que la consola, usamos la librería streamlit. Luego, para correr la simulación, usted deberá hacer:

```
streamlit run src/simulate.py
```

### 3 Implementación

Para la implementación, creamos una clase `Serial Servers Sim`. Al instanciar esta clase le pasamos todas las distribuciones, dígame la de la llegada de clientes y las relativas al tiempo de servicio de cada servidor. Esta clase tiene un atributo `timeline`, que no será otra cosa sino la lista de eventos de la simulación. Además esta clase tiene una tabla donde almacenara los datos generados por la simulación. En tal tabla, a cada cliente corresponde una fila y a cada servidor una columna, y las entradas de la tabla tienen la forma de tuplas donde aparecen el tiempo de llegada del cliente a la cola del servidor, el momento en que comenzó a ser atendido y el momento en que dejó el servidor.

Para realizar la simulación, se requiere un parámetro `time` (el tiempo a simular). Luego, usando la distribución que haya sido provista para simular los intervalos de tiempo entre las llegadas de clientes, obtenemos una lista de números que siga tal distribución y cuya suma no sobrepase `time`. Tal lista corresponderá a los intervalos de tiempo de llegadas de clientes hasta el tiempo `time`, a partir del cual asumimos no llega ningún nuevo cliente. Teniendo así las llegadas que ocurrirán, podemos construir la tabla, colocando una columna por cada cliente. Los eventos en `timeline` son tuplas, que como primera componente tienen el momento de ocurrencia del evento, como segunda la naturaleza del evento "ARRIVAL", "BEGIN SERVICE" o "END SERVICE", como tercera el id del cliente y como cuarto el id del servidor. `Timeline` es una cola de prioridad, que ordena los eventos por tiempo de ocurrencia en orden creciente.

La clase `Serial Servers Sim`, cuenta además con una lista de objetos de tipo `Service`. Cada objeto de tipo `Service` maneja la información relacionada con uno de los servidores en particular. Creímos necesario modelar a los servidores de esta manera, puesto que nos sería muy difícil extraer de la tabla del post procesado información relativa al tamaño de la cola de cada servidor. Además un servidor debe manejar cierta lógica para manejar su cola de clientes, y dado que el número de servidores en cualquier simulación con sentido va a ser muchísimo menor que el número de clientes, el costo computacional de levantar una instancia por cada servidor no representara un cuello de botella. Por razones opuestas decidimos no modelar a los clientes como una clase, y extraer toda la información relativa a ellos en el post procesado de la tabla que generamos durante la simulación. O sea, tales razones opuestas serían: el poco valor que tiene para cualquier estudio diferenciar la

información de lo que ocurrió a un cliente en particular, pues la masa de los clientes es tratada como homogénea e infinita; la enorme cantidad de clientes que provocarían un costo computacional elevado al tener que instanciar una clase para cada uno de ellos; y además la ausencia de una lógica a modelar para los clientes, ellos tan solo fluyen, como el agua. Por supuesto si se quisieran hacer simulaciones donde, por ejemplo, los clientes se asustaran en caso de encontrar una cola muy larga y se marcharan, comenzaría a tener sentido crear una abstracción para ellos.

Cada instancia de tipo Service, cuenta con métodos para manejar cada uno de los eventos. Al ser llamados, le es pasado a la instancia el tiempo en que estamos en la ejecución, de manera que puede actualizar sus datos, haciendo uso de una función interna suya `update time`.

La clase Serial Servers Sim además, tiene métodos para manejar cada uno de los eventos y un método para el post procesado, que consiste meramente en iterar por cada fila y cada columna de la tabla para recuperar los datos de interés.

## 4 Modelo Matemático

El modelaje matemático es el de un sistema de colas donde asumimos población infinita. Asumimos llegada de clientes a intervalos de tiempo de distribución exponencial. Si bien nuestro sistema tolera que se coloque cualquier otra distribución para la llegada de clientes, verdaderamente no creemos que tenga sentido usar otra. Los servidores tienen un tiempo de servicio que distribuye acorde a como el usuario de la simulación establezca.

### 4.1 Supuestos y Restricciones

Asumimos que automáticamente cuando un servidor termina de atender a un cliente, pasa al siguiente en la cola. Asumimos que al cliente no le toma tiempo moverse de una cola a otra. Luego, esta simulación es más cercana a lo que ocurre en sistemas similares computarizados.