

# Proyecto Eventos Discretos:

Abel Antonio Cruz Suárez C-411

November 21, 2021

## Orden del problema

La cocina de Kojo es uno de los puestos de comida rápida en un centro comercial. El centro comercial está abierto entre las 10 : 00 am y las 9 : 00 pm cada día. En este lugar se sirven dos tipos de productos: sándwiches y sushi. Para los objetivos de este proyecto se asumirá que existen solo dos tipos de consumidores: unos consumen solo sándwiches y los otros consumen solo productos de la gama del sushi. En Kojo hay dos períodos de hora pico durante un día de trabajo; uno entre las 11 : 30 am y la 1 : 30 pm, y el otro entre las 5 : 00 pm y las 7 : 00 pm. El intervalo de tiempo entre el arribo de un consumidor y el de otro no es homogéneo pero, por conveniencia, se asumirá que es homogéneo. El intervalo de tiempo de los segmentos homogéneos, distribuye de forma exponencial. Actualmente dos empleados trabajan todo el día preparando sándwiches y sushi para los consumidores. El tiempo de preparación depende del producto en cuestión. Estos distribuyen de forma uniforme, en un rango de 3 a 5 minutos para la preparación de sándwiches y entre 5 y 8 minutos para la preparación de sushi. El administrador de Kojo está muy feliz con el negocio, pero ha estado recibiendo quejas de los consumidores por la demora de sus peticiones. Él está interesado en explorar algunas opciones de distribución del personal para reducir el número de quejas. Su interés está centrado en comparar la situación actual con una opción alternativa donde se emplea un tercer empleado durante los períodos más ocupados. La medida del desempeño de estas opciones estará dada por el porcentaje de consumidores que espera más de 5 minutos por un servicio durante el curso de un día de trabajo. Se desea obtener el porcentaje de consumidores que esperan más de 5 minutos cuando solo dos empleados están trabajando y este mismo dato agregando un empleado en las horas pico.

## Principales ideas seguidas para la simulación del problema

La idea principal que se desarrolla en el trabajo es un flujo donde se tienen 2 eventos, la llegada de clientes al sistema y la salida de los mismos. En nuestro caso debemos agregar además que en las horas pico podemos tener o no una cierta cantidad de trabajadores extra. Para este problema nos basamos en una línea temporal donde vamos avanzando en el tiempo y en dependencia del menor tiempo entre una llegada y las posibles salida, analizamos un evento u otro.

El flujo sería el siguiente:

- Inicializamos las variables contadoras
  - $N_a$ : Cantidad de llegadas totales al sistema.
  - $N_d$ : Cantidad de salidas totales del sistema.
  - *people\_in\_syst*: Cantidad de clientes actualmente en el sistema.
  - *time\_line*: Tiempo actual del sistema (línea de tiempo).

- *Arrivals*: Diccionario donde se indica el cliente  $i$  en el tiempo en que llegó.
- *Departures*: Diccionario donde se indica el cliente  $i$  en el tiempo en que se fue.
- $t\_d$ : Lista donde se llevan las próximas salidas de los clientes que se están atendiendo en un servicio.
- $t\_a$ : Próxima llegada de un cliente que se va a atender.

- Evento de llegada:

Un evento de llegada se atiende si el menor tiempo entre  $t\_a$  y el menor de los tiempos de salidas en  $t\_d$  es  $t\_a$  y este a su vez es menor que el tiempo de simulación. Una vez en esta fase aumentamos la cantidad de clientes en el sistema, la cantidad de clientes que han llegado en total, la línea temporal y registramos la llegada del cliente en el diccionario de llegadas.

Luego procedemos a generar la llegada del próximo cliente( $i + 1$ ) y este será el nuevo valor de  $t\_a$ . Como estamos analizando la llegada de un cliente  $i$  debemos saber que pedido realiza. Para esto tenemos el método *Order* que no da el valor de una variable aleatoria con distribución de Bernoulli.

Como ya tenemos el cliente y la orden que quiere realizar pasamos a introducirlo en la cola. Para esto se usa el método *UpdateQueue*. En esta función utilizamos *SelectLine* para determinar en que cola se debe colocar un cliente. Con el método *SelectLine* si nos encontramos en una hora pico entonces el cliente podrá seleccionar la menor cola de los servicios. En caso de no ser una hora pico se selecciona solo dentro de las dos primeras. Una vez que tenemos la cola seleccionada y el servicio, si es un cliente que llega a un servicio vacío se calcula su tiempo de salida, que dependerá del pedido. Si el servicio está atendiendo otros clientes entonces se encola junto con su pedido.

- Evento de Salida:

Para atender un evento de salida debemos tener en cuenta que tenemos  $n$  servicios en los que se encuentran los respectivos clientes que se están atendiendo. Entonces de todos esos clientes se atenderá la salida del que tenga el menor tiempo almacenado en la lista  $t\_d$ , si este tiempo es menor que la próxima salida a atender y que el tiempo de la simulación. Una vez en esta fase aumentamos la cantidad de clientes que sale del sistema, la cantidad de clientes que han salido en total, la línea temporal y registramos la salida del cliente en el diccionario de salidas. Para saber de que servicio sale el cliente que se está atendiendo utilizamos el método *NextOut*. Si el cliente al salir era el único que quedaba en el servicio pasamos a poner a ese trabajador en estado de libre(indicado con el valor de infinito), si no era el último, entonces generamos la salida del próximo cliente( $i + 1$ ) en

la cola de dicho servicio, que dependerá del pedido que este haya querido cuando llegó. Y en cualquier caso procedemos a despachar al cliente  $i$ .

Para saber si este cliente estuvo más de 5 minutos esperando se resta el instante de tiempo en que partió menos el instante de tiempo en que llegó.

- Evento de Cierre:

En esta fase se está cuando un cliente fuera de tiempo llega al servicio. Entonces procedemos solo a atender las salidas de los clientes que queden en el sistema.

Al igual que en los otros eventos actualizamos las variables contadoras, determinamos quién es el próximo cliente en abandonar el servicio y procedemos a atender dicho evento de salida.

Como dato importante destacar que se utiliza una generación de variables con función de distribución uniforme para la elaboración de los pedidos, para las llegadas se utiliza una con función de distribución exponencial.

La simulación la realizamos 1000 veces para tres valores de  $\lambda$  distintos:  $\lambda = 1/5, 1/9, 1/13$ . En total se realizan 6000 simulaciones, una por cada  $\lambda$ , y cuando hay ayudantes y cuando no.

## Modelo de Simulación de Eventos Discretos desarrollado para resolver el problema

Para este problema utilizamos un modelo de simulación de servidores en paralelo. Aunque se nos habla de analizar el caso donde haya un trabajador más en las horas pico(3 servidores en serie) y cuando no(2 servidores en serie). Aquí se desarrolla un modelo para  $n$  servidores en paralelo, permitiéndonos analizar cuando se utilizan más de un trabajador extra en las horas pico. Se permiten también el uso de diferentes valores de  $\lambda$  para las llegadas de los clientes dentro y fuera de las horas pico.

## Consideraciones obtenidas a partir de la ejecución de las simulaciones del problema

Para obtener el porcentaje de personas que esperan más de 5 minutos, utilizamos la clase *Statistics*. En esta clase luego de cada simulación guardamos en una variable los clientes que se atendieron más los que se habían atendido en simulaciones anteriores(*Total\_clientes*) y de los que se atendieron guardamos en otra variable cuantos esperaron más de 5 minutos más los que esperaron más de 5 minutos de anteriores simulaciones(*Parte\_5\_mins*). Al concluir un ciclo de simulación(1000), para obtener los porcentajes se divide  $\frac{Parte\_5\_mins}{Total\_clientes} * 100$  y esto nos dará

en un ciclo de 1000 simulaciones el % de personas que esperó más de 5 minutos. Para el análisis de los datos usamos valores del  $\lambda = \frac{1}{2}$  para la llegada de los clientes en las horas pico. Lo que implica que en estas horas llegarán los clientes, de forma consecutiva, más rápido.

Como Conclusión obtenemos que usando valores de  $\lambda$  menores para los arribos, la diferencia entre la cantidad de personas que esperan más de 5 minutos cuando hay trabajadores extra en las horas pico es mayor que para valores menores. En el caso que se generó se comprueba que como promedio alrededor de 24 personas más esperan más de 5 minutos cuando no hay trabajadores extra. Este valor es menor a medida que usamos valores de *lambda* mayores para los arribos. Esto se debe a que los arribos ocurren de manera más seguida y por tanto se pueden realizar mayor cantidad de pedidos sin que lleguen más clientes y aumenten la cola. Para *lambda\_arribo* =  $\frac{1}{9}$  se obtiene que en promedio la diferencia es de alrededor de 18 clientes y para *lambda\_arribo* =  $\frac{1}{13}$  de 16.

Como se puede comprobar esta no constituye una gran diferencia. Pues de manera general siguen esperando más de 5 minutos la gran mayoría de los clientes. Esto lo comprobamos con que para *lambda\_arribo* =  $\frac{1}{5}$  el 78.54296871118707% de los clientes esperó más de 5 minutos cuando hay ayudantes, en comparación con el 90.09050961819506% cuando no los hay. Es decir que solo disminuye el porciento de los clientes que esperan más de 5 cuando hay ayudante en 11.547540907007985%. Cuando *lambda\_arribo* =  $\frac{1}{9}$  el 77.22964172426042% de los clientes esperó más de 5 minutos cuando hay ayudantes, en comparación con el 88.15431117758874% cuando no los hay. Lo que significa una disminución de un 10.92466945332832%. Para *lambda\_arribo* =  $\frac{1}{13}$  el 78.2771063233491% de los clientes esperó más de 5 minutos cuando hay ayudantes, en comparación con el 89.20146265018283% cuando no los hay. Lo que significa una disminución del 10.924356326833731%.

Como se pudo apreciar en los datos obtenidos con la presencia de un ayudante de más, solo hay una disminución de alrededor de un 10% en los clientes que esperan más de 5 minutos. Lo que no puede considerarse una gran diferencia. Pues para el caso donde *lambda\_arribo* =  $\frac{1}{5}$  hay en total 201286 clientes que esperan más de 5 minutos de un total de 201857. De forma igual ocurre en los otros casos de *lambda\_arribo*.

Si bien la presencia de 1 trabajador más no hace que disminuya mucho el tiempo de personas que esperan, si aligera la carga de los otros cocineros, pues para el caso de *lambda\_arribo* =  $\frac{1}{5}$  este tercer cocinero atendió en 17.031984340689366% de los clientes, cuando *lambda\_arribo* =  $\frac{1}{9}$  atendió el 20.1897010525791% y para el caso de *lambda\_arribo* =  $\frac{1}{13}$  atendió el 21.926782273603084%

Por tanto si se quiere disminuir más el porciento de clientes que esperan más de 5 minutos se recomienda emplear más de 1 ayudante para las horas pico. Pero si se desea aligerar la carga en cocina si se pudiera emplear otro cocinero pues atiende un porciento significativo de los clientes.

En este trabajo se da la posibilidad de realizar las simulaciones con más de 1 ayudante. Para esos casos se encontraron mejoras pero su discusión va más allá de lo que desea saber el empleador.

## **Nota**

Se adjunta un documento .txt con la salida de la simulación realizada y descrita aquí.

## **Especificaciones del código**

- Versión de Python utilizada: 3.8.10 64 bit.
- Bibliotecas empleadas: fractions, numpy, sys.