

# Kojo's Kitchen

Daniel de la Osa Fernandez  
c412

March 24, 2019

## 1 Análisis de Kojos Kitchen

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 sólo dos tipos de consumidores: unos consumen sólo sándwiches y los otros consumen sólo 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, distribuyen de forma exponencial con el parámetro  $\lambda$  tal que como media los intervalos de llegadas entre clientes son la siguiente:

Período	Intervalo de llegadas $\lambda$ (mins)
10:00-11:30	16
11:30-13:30	2
13:30-17:00	13
17:00-19:00	3
19:00-21:00	10

Actualmente dos empleados trabajan todo el día preparando sándwiches y sushis 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 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.

## 2 Ideas seguidas para la solución

Este problema es muy similar al problema de atender a clientes con  $n$  servidores en paralelo con la diferencia de que todo el tiempo no están disponibles estos servidores, en nuestro problema los cocineros. Los clientes comenzaran a llegar a la cafetería a pedir sándwich o sushi con siguiendo una distribución exponencial como indica el ejercicio.

Cuando un cliente llega uno de los cocineros lo atiende lo que equivale a generar el tiempo de salida del cliente del establecimiento. Esto se logra generando una variable aleatoria con distribución uniforme( $a$ ,  $b$ ) en dependencia si lo que quiere es sándwich o sushi, para saber que es lo que quiere el cliente se genera un v.a con distribución bernoulli con  $p=0,5$ . Esto se podrá variar en dependencia si las personas piden mas un producto u otro.

Otro caso será cuando el cliente llegue y no hay ningún cocinero disponible en este caso se coloca en una cola y hasta que llegue el tiempo de salida de alguno de los que se están atendiendo y entonces se le da su tiempo de salida, siempre que sea el primero de la cola sino solo se acerca más a ser atendido pero sigue encolado. Se definió el tiempo de espera del cliente como el tiempo en que salió de la cola menos el tiempo en el que llegó al establecimiento.

Sea  $T$  el tiempo en el que el establecimiento cierra, todos los clientes que estén en cola serán atendidos y los clientes que lleguen pasada esa hora no se dejará que entren; así el sistema podrá atender a los que falta y después terminar.

Para el análisis que se pide la idea es trabajar con la media del tiempo de demora de los clientes en varios días de simulación de la cocina de Kujo. Esta cantidad de días la puede definir el usuario pero de no ser así se simulan 30 días inicialmente y luego se utiliza una fórmula estadística para saber si con esa cantidad de días es suficiente para estimar el parámetro que se quiere con un error previamente definido en este caso se tomo error de 1 minuto. La fórmula utilizada fue (formula). Donde  $d=1$ ,  $S$  es la raíz del estimador puntual para la varianza y  $n = k$  la cantidad de días simulados hasta ese momento.

## 3 Modelo

Aquí se presentará el modelo seguido para lograr la simulación basada en eventos discretos donde las variables y los eventos toman un papel fundamental.

### 3.1 Estados

- Llegada de un cliente a la cocina de Kujo
- Partida de un cliente que esta siendo atendido por el cocinero  $i$

### 3.2 Variables

- $t$  : Tiempo actual en que se encuentra el sistema.

- $T$  : Tiempo en el cual cierra la tienda.
- $n$  : Cantidad de clientes totales que hay el sentido.
- $nA$ : Cantidad de clientes que han llegado al sistema.
- $cooks$ : Número de cocineros.
- $ta$ : Tiempo de llegada.
- $ti(i)$ : Tiempo en que el cocinero  $i$  se desocupa.
- $D(i)$ : Tiempo de partida del cliente  $i$ .
- $TimeInWait(i)$ : Tiempo en que sale el cliente  $i$  de la cola.
- $SS = (n, i1, i2, ..., iN)$  ( $n$  clientes en el sistema, con  $i$  sub  $j$  atendido por el cocinero  $j$  en este caso solo  $j = (1, 2, 3)$ . Si  $n > N$  los  $n - N$  restantes están en cola). Si  $S = (0)$  significa que el sistema está vacío. Si para algun  $i$  si  $S[i] = 0$  significa que el cocinero  $i$  esta disponible.

### 3.3 Inicialización

- $t0 = 600$  (Esto corresponde a las 10am en minutos).
- $T = 1260$  (Equivale al tiempo de cierre)
- $ti = [infinity] * cooks$
- $ta = t0$
- $demand\ interval = False$  (Variable para indicar el cambio de intervalo de uno normal a otro critico importante cundo se simula con tres cocineros)

### 3.4 Llegada de Cliente

El primer caso que queremos ver es el arribo al sistema de un cliente esto siempre va a ejecutarse siempre que sea el tiempo mas cercano al actual( $t$ ) y que sea posible que llegue alguien es decir que  $ta < T$ .

Al ocurrir esto se aumenta  $nA$  así como  $n$  ya que llego alguien nuevo al sistema. Luego hay tres opciones el sistema está vacío, hay al menos un servidor vacío o todos están llenos.

Si ocurre lo primero hacer  $SS = (1, nA, 0, 0)$  (poner al cliente en el 1er cocinero) y generar su tiempo de salida y hacer  $ti(0) = t + \text{valor generado}$

Si ocurre lo segundo se pone el cliente en el lugar del cocinero disponible. Tener en cuenta que si estamos en el caso de tres cocineros hay que revisar en que intervalo estoy, ya que si no es el crítico tienen que haber al menos dos disponibles ya que un cocinero va dejar de trabajar. De nuevo se genera el tiempo de salida del cliente que se comenzó a atender.

Si ocurre el tercero pues se encola al cliente al final de  $SS$  y se aumenta  $SS[0]$  en uno.

Luego se genera el próximo arribo haciendo  $ta = t + \text{valor generado}$

### 3.5 Partida de un cliente del cocinero i

En este caso va a entrar si el tiempo de terminar que marca  $ti[i]$  es el próximo en la lista de eventos en el tiempo es decir si es el menor de todos los posibles tiempos a moverse.

Nos movemos a ese tiempo haciendo  $t=ti[i]$  disminuimos en 1 a  $n$  y hacemos tambien guardamos el tiempo de salida para el cliente que estaba siendo atendido por él.

Si el solo quedaba ese cliente en el sistema hacemos  $SS=[0]$  y  $ti[i]=\text{infinito}$ . Si no hay clientes en cola hacer cero la posición de  $SS$  correspondiente a ese cliente y poner  $ti[i]=\text{infinito}$

Si existen clientes en cola simplemente poner el próximo de la cola en la posición de  $SS$  del cocinero que terminó y generar la salida de ese cliente para actualizar  $ti[i]$  al igual que se hace en el caso de entrada. Tener en cuenta que si es para tres cocineros pasa lo mismo de que si estas en un intervalos que no es de demanda entonces no se coloca ese cocinero a trabajar simplemente se hace cero en  $SS$  y se pone a infinito  $ti[i]$ .

### 3.6 Caso cierre

Si no existe nadie en el sistema es decir  $n$  es igual a cero o  $SS=(0)$  y el próximo tiempo a moverme es mayor que  $T$  pues termine el proceso de un día de simulación

## 4 Resultados

Con este proyecto se logró mediante una simulación basada en eventos discretos describir el funcionamiento de la cocina de Kujo. La aplicación nos permite generar simulaciones para la cantidad de días escogidos con intervalos de llegada de los clientes también modificables y en todos los casos el poner un nuevo cocinero resulto factible en disminuir el por ciento de clientes insatisfechos. También jugar con los parámetros de entrada pueda dar otras informaciones como son la media de espera o la necesidad de que aunque con tres empleados es mejor para un flujo determinado puede ser aun insuficiente para lograr un mejor servicio a clientes. Ya que aunque esperan menos los clientes siguen esperando mucho tiempo para algún flujo de llegada  $x$ . También la aplicación permite la escalabilidad del negocio probando con nuevas configuraciones antes de crecer y así aprovechar al maximo la inversión. Finalmente mostramos una salida del programa para 100 días con los lambdas por defecto.

## 5 Anexos

Corrida.png

Figure 1: